The Dissertation Committee for Bo Lu

certifies that this is the approved version of the following dissertation:

Iteratively Coupled Reservoir Simulation for Multiphase Flow in Porous Media

Committee:

Mary F. Wheeler, Supervisor

Clint Dawson

Ekwere J. Peters

Kamy Sepehrnoori

Steven L. Bryant

Iteratively Coupled Reservoir Simulation for Multiphase Flow in Porous Media

by

Bo Lu, B.S., M.S

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May 2008

To my parents,
Wencai Lu and Chengfang Tian,
and to my wife, Ping

Acknowledgments

First I would like to thank my supervisor, Professor Mary F. Wheeler, for her support and supervision. She has guided my studies and encouraged my work in computational and applied science and engineering. She has not only mentored me academically as a researcher; she has also taught me valuable life lessons.

I would like to thank Dr. John Wheeler, who develop the framework of IPARS. All the work done in this dissertation is under IPARS. The specialty of IPARS benefits this research in various aspects.

Special thanks are due to Dr. Xiuli Gai. She is always gracious to help me with technical problems, and she shares generously of her own experiences. Thanks are also due to Dr. Tareq M. Alshaalan, who has been a collaborator and readily provides valuable suggestions and explanations. I would also like to thank Klaus Stuben and Tanja Clees from Fraunholfer Institute SCAI, Alexander Zekulin and Kirk Jordan from IBM, for their assisting on linear solver and HPC studies.

I would also like to thank Dr. Hector Klie, Dr. Adolfo Rodriguez, Dr. Shuyu Sun, Sunil Thomas, and Horacio Florez. Our cooperation and ongoing discussion have provided insight and contributed greatly to the work of this dissertation. Special thanks are also due to Andrew Pacetti for help with revisions.

My gratitude also goes to the members of my committee, Professor Clint Dawson, Professor Ekwere J. Peters, Professor Kamy Sepehrnoori, and Professor Steven

L. Bryant, for reviewing this dissertation and providing helpful suggestions and comments.

I would like to thank all my colleagues and the members of the Center for Subsurface Modeling (CSM) for their help with this work. Thanks are also due to the Department of Energy and the Industrial Affiliates of CSM for their continued support.

Last, but by not means least, I want to thank my parents for their unconditional love and support. And to my dear wife, Ping, who is always with me whenever I need her, sharing all of her life with me, and without whose encouragement and love this dissertation could not have been accomplished, thank you.

Bo Lu

The University of Texas at Austin

May 2008

Iteratively Coupled Reservoir Simulation for Multiphase Flow in Porous Media

Publication No. _____

Bo Lu, Ph.D.
The University of Texas at Austin, 2008

Supervisor: Mary F. Wheeler

Fully implicit and IMPES are two primary reservoir simulation schemes that are currently used widely. However, neither of them is sufficiently accurate or efficient, given the increasing size and degree of complexity of highly heterogeneous reservoirs. In this dissertation, an iterative coupling approach is proposed and developed to solve multiphase flow problems targeting the efficient, robust and accurate simulation of the hydrocarbon recovery process.

In the iterative coupling approach, the pressure equation is solved implicitly, followed by the saturation equation, which is solved semi-implicitly. These two stages are iteratively coupled at the end of each time step by evaluating material

balance, both locally and globally, to check the convergence of each iteration. Additional iterations are conducted, if necessary; otherwise the simulation proceeds to the next time step. Several numerical techniques are incorporated to speed up the program convergence and cut down the number of iterations per time step, thus greatly improving iterative model performance. The iterative air-water model, the oil-water model, and the black oil model are all developed in this work.

Several numerical examples have been tested using the iterative approach, the fully implicit method, and the IMPES method. Results show that with the iterative method, about 20%-40% of simulation time is saved when compared to the fully implicit method with similar accuracy. As compared to the IMPES method, the iterative method shows better stability, allowing larger time steps in simulation. The iterative method also produces better mass balance than IMPES over the same time.

The iterative method is developed for parallel implementation, and several test cases have been run on parallel clusters with large numbers of processors. Good parallel scalability enables the iterative method to solve large problems with millions of elements and highly heterogeneous reservoir properties.

Linear solvers take the greatest portion of CPU time in reservoir simulations. This dissertation investigates advanced linear solvers for high performance computers (HPC) for reservoir simulation. Their performance is compared and discussed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In reservoir simulation, computational cost (CPU time) minimization and accuracy are two contradictory goals that are difficult to obtain simultaneously. Several attempts have been made and more are continuing to pursue the development of reservoir simulators that are both accurate and efficient. Reservoir simulators require accuracy to model the performance of reservoirs of large size and high heterogeneity, with small material balance errors and small solution oscillations. Both high performance computers and fast solvers are required to ensure efficiency. The recent increase in the availability of parallel computational technologies provides significant facility support, and developments in robust linear solvers, grid upscaling/refinement, and uncertainty have also increased efficiency. In this dissertation, iteratively coupled reservoir models are developed and implemented with parallel computation, resulting in a new scheme that is targeted to solving the aforementioned problems.

1.1.1 Problem Statement

Multiphase Flow Simulation

Reservoir simulation with a compositional model is beyond the scope of this dissertation. Unless stated otherwise, multiphase flow in this dissertation refers to two-phase and black oil flow.

Reservoir simulation for multiphase flow is critically important to understanding fluid flow in porous media and the changes of reservoir properties during the hydrocarbon recovery process; to suggest an optimal production plan for a given reservoir; to make new drilling plans for mature reservoirs; and to adopt efficient techniques for secondary and tertiary oil recovery. Physical and numerical models of reservoir simulation have been in development for many years, from simple 1D single phase flow models to complicated 3D compositional models. At the same time, however, problems have become larger and larger. Several solution methods have been investigated to solve the equation systems of existing numerical models. These approaches include the fully implicit method (FIM), the implicit pressure explicit saturation (IMPES) method, and the sequential method *etc.* Each of these approaches has advantages and disadvantages. The FIM is unconditionally stable and it is the most accurate method. Sometimes, however, it is difficult to implement and computationally costly. The IMPES method is very efficient, and it has been used successfully on several problems. However, it exhibits oscillation and unacceptably small time steps for problems with particularly complicated features, such as countercurrent flow problem. The sequential method usually has better performance than the FIM and the IMPES. For reservoir problems with low heterogeneity, the sequential method takes 1/2 or 1/3 simulation time of the FIM. However, the

sequential method has difficulty to solve reservoir problem with quickly changing capillary pressure. For fractured reservoir, the FIM is the first choice.

The above three approaches, in addition to several other solution attempts, have already been discussed broadly. They have also been implemented in several simulators with enhanced functionalities. However, a reservoir model with a favorable combination of the advantages of current methods is still a goal of research. The formulation of a comprehensive model of multiphase flow that is stable, efficient, and able to handle most general reservoir problems is still an open problem. Additionally, this desired model should have the ability to conduct full-field reservoir simulations with multiple processors, handling reservoirs with millions of elements. The research focus of this dissertation is the formulation of a new solution methodology to overcome the problems stated here.

HPC Reservoir Simulation

In recent decades, computational costs have hindered, to some degree, the development of new ideas and technologies in reservoir simulation. Additionally, the rapid development of petroleum technologies increasingly requires that reservoir simulators handle larger and more complex problems. Reservoir simulators are now required to handle reservoirs with very fine meshes, millions of grid blocks, and high heterogeneity. The idea of a *next generation reservoir simulator* is proposed in this dissertation to satisfy these requirements. The next generation simulator should have the ability to deal with huge and complicated reservoirs. At the same time, simulations should have high efficiency in solving problems within a short time. All of these requirements point out needs in the computation facilities. The construction of parallel clusters provides a potential solution to this requirement.

With parallel computation, one large problem can be divided into smaller subproblems, and these can then be solved with multiple processors simultaneously. Each processor only deals with its local data in one time step. The solution of subproblems will be integrated into the global solution at the end of each time step. The efficient design and implementation of reservoir simulators, including parallel scalability under parallel computation environments, are new issues currently under investigation.

Substantial effort has been expanded in the research of linear solvers and preconditioners for reservoir simulation. The particular architectural characteristics of high performance computing (HPC) also require new linear solvers that are parallel-oriented. These solvers should be able to solve local problems and deliver global solutions while maintaining efficient and accurate parallel communication during linear solving.

HPC reservoir simulation is an art of design and optimization. It is far beyond simply submitting data and running programs. It is machine and problem dependent. Research is still ongoing to optimize parallel simulation and to utilize computing resources effectively.

1.1.2 Research Objective

The main objective of this work is to develop an iteratively coupled multiphase flow model that is accurate and efficient in simulating the hydrocarbon recovery process. Such a model is required for history-matching, optimal control and improved production planning for reservoir engineering. This model must also be applicable to most general parallel computing environments with high scalability in order to handle large scale reservoir problems. Advanced solvers and numerical techniques

will be investigated with this iterative model. The iterative method will combine the advantages of several other methods while maintaining stable performance and reasonably sized time steps. Additionally, the iterative approach is flexible enough to couple different physical models and different numerical algorithms within one simulation framework, without additional computation cost. Model results should also have small material balance errors.

Some outlined objectives are listed as follows:

- Develop an iteratively coupled model for the two-phase oil/water problem
- Develop an iteratively coupled model for the two-phase air/water problem
- Extend the iterative coupling approach to the black oil model
- Investigate the variety of numerical techniques that can be applied to the iterative model to improve the solution and accelerate the simulation
- Investigate the parallel implementation of the iterative model with good scalability
- Investigate linear solvers for chosen reference pressure equations and adaptive choices of convergence tolerances
- Investigate linear solvers for HPC simulation
- Study optimal parallel simulations.

1.2 Literature Review

1.2.1 Black Oil Model

Reservoir simulation was an early development in the science of reservoir engineering. Early research on reservoir simulation was mostly on small scale reservoir with low heterogeneity. Early computing technology in the 1960's facilitated both the development of the black oil model and mathematical research into solvers for large systems of equation. Several people such as Douglas, Peaceman, and Rachford *et al.* contribute to the pioneer work on reservoir simulation [6, 36, 86, 87]. Since then, the three-phase black oil fluid treatment has remained the standard for reservoir simulators [23].

The 1970's saw the derivation of the conventional formulation of the black oil model and its broad use in the simulation field [6, 13, 66, 86, 125]. The main attraction of the black oil model at that time was that it accounted for the four basic oil recovery mechanisms: (1) fluid expansion, (2) displacement, (3) gravity drainage, and (4) capillary imbibition [23]. Cook proposed a method for extending the black oil model to include gas injection with non-equilibrium gases for represent into more complex compositional effects, including when three or more components are required [29]; Spivak and Dixon then made new modifications to the black oil formulation to account for volatile oil and gas condensate fluids, where the oil component may exist in the gas phase [106].

In the 1980s, the black oil model was applied to more complicated, real-world conditions. Dimitrie *et al.* described an approach for simulating three-phase flow in a fractured reservoir that was based on the dual porosity concept with certain modifications to the 3D black oil model [35]. Whitson *et al.* modified the black oil

model with simplified compositional PVT formulations to solve the full-field simulation problems of several North Sea reservoirs that contained near-critical fluids with compositions that varied areally and with depth [124]. The modified black oil model included four components: non-volatile surface oil, volatile surface oil, surface hydrocarbon gas, and injection gas. Simulation of a heavy oil reservoir with the black oil model had been addressed by Huan [60, 61], in which he proposed the *flash* approach. Several papers treated black oil simulation as a special case of compositional simulation [56, 85, 117, 127], and the behaviour of the gas phase was investigated specifically [44, 48]. In particular, Trangenstein and Bell provided a detailed mathematical structure of the black oil model [113], including thermodynamic equilibrium, an equation of state (EOS) and component conservation equations. It should be noted that the black oil models developed during this decade were beginning to realize their potential to simulate full-field reservoir problems.

Since the 1990s, most research has focused on the various black oil model solution methodologies [14, 25, 28, 51, 113, 127]. Several general purpose black oil models have been developed to handle comprehensive reservoir problems [24, 55, 109]. Tan and Kaiogerakis described a reservoir simulator with an automatic history matching capability [109]. Ganzer and Heinemann developed a multipurpose reservoir simulator to solve a wide range of reservoir problems, including three-phase black oil depletion, miscible displacement of multi-component fluid [55]. The black oil model has also been used together with other models, such as the compositional model, to deal with particular reservoir conditions. Fevang, Singh and Whitson discussed the guidelines for choosing different models for volatile oil and gas-condensate reservoirs [46]. They discovered that the black oil model can be used for most depletion cases with proper PVT data; however, it is not suitable in most cases of

gas injection. In the last decade, much effort has also been expended in solving the black oil problem using powerful parallel computation technologies [2, 67, 81].

Streamline-based methods and domain decomposition are shown to be two efficient ways to solve large-scale black oil problems [3, 10, 78, 122]. Agarwal *et al.* presents a method for history matching watercut data with a streamline simulation that captures all the pertinent physics [2]. 1D flow along streamlines is assumed and the production water rate is investigated for its sensitivity to permeability changes. Compressibility and gravity have been detected to have a significant influence on this sensitivity. In 2003, Berenblyum added the capillary pressure and gravity factors into the streamline simulation for the black oil systems [10], showing that the capillary forces stabilize the displacement front and increase oil production. In 2005, a rigorous compressible streamline formulation was developed by Cheng *et al.*, which extended streamline applications from incompressible flow to compressible flow [20]. In their model, fluid density along the streamline was updated according to changes in phase pressure; the calculation of saturation and the advantages of the streamline method were preserved.

The black oil model with thermal functionality is another developing topic in reservoir simulation. It is becoming more and more important together with the enhancement of oil recovery, especially with regard to steam injection [9, 19].

1.2.2 Solution Methodology

There are three common approaches for solving the differential equation systems in reservoir simulation. These three methods are the fully implicit method (FIM), the implicit pressure explicit saturation (IMPES), and the sequential method.

The FIM is the most widely used solution approach of reservoir simulation. In

an implicit formulation, all terms in the system, such as transmissibility, well term, and capillary pressure, are treated implicitly, and all of the variables are solved simultaneously. This approach is unconditionally stable and allows large time steps. Additionally, the FIM is shown to be accurate with several numerical examples, and it is described in a number of papers [6, 28, 45, 86, 94, 96, 109, 118]. However, FIM has a complicated numerical structure that makes it generally difficult to solve, especially for large problems. The implementation of FIM is also labor-intensive. As a result, attention has moved on to other methods.

IMPES is the fastest approach on a per-time step basis. It was well discussed by Coats in 1968 in a description of a numerical black oil model [25]. Fagin and Steward even discussed an IMPES model earlier in 1966, with an application to a 2D three-phase reservoir [43]. Even though they did not use the now-common acronym IMPES, they did sum all the mass equations to form a pressure equation, and then they solved the saturation equation explicitly. For a fluid flow system, the basic goal of the IMPES method is the elimination of differences in non-pressure variables to obtain a single pressure equation; this pressure equation is solved implicitly, while all other variables in the pressure equation are treated explicitly [25]. Once the pressure has been obtained, other unknowns are solved explicitly, in sequence. Further developments and extensions of research into the IMPES approach have been realized [23, 117, 127], and many IMPES-based methods have been proposed [14, 15, 25], both for the black oil model and for the compositional model. However, all IMPES methods exhibit stability problems and restricted time steps [26, 27]. Chen *et al.* discussed an improved IMPES method for two-phase flow, in which different time-stepping for pressure and saturation is applied [17]. This approach was based on the assumption that pressure changed less rapidly than saturation

with respect to time. Within a large pressure step, multiple smaller saturation steps were realized by an adaptive time step control strategy. The same approach has been applied to the IMPES model in IPARS [78].

In most IMPES-based methods, time step controls have been set to limit the maximum time step size. Details regarding stability for IMPES method will be addressed later in this dissertation. The IMPES-based methods also face difficulty in handling high flow rates, such as in areas near wellbores. Under most circumstances, the IMPES method must be used together with other methods.

The sequential method was first applied to the black oil model by Watts in 1985 [117], using the same pressure formulation as in the IMPES method, except that it was followed by an implicit saturation formula. It had been shown that the sequential solution is computationally more efficient than the FIM and very stable. It permits the use of a more stable algorithm when simulating irregular grids. However, the sequential method requires additional memory for the temporary storage of previous solutions, making it more costly compared to other approaches [1].

The Adaptive Implicit Method (AIM) is an attempt to combine the advantages of the fully implicit and the IMPES methods by solving the problem implicitly in some local areas, such as near wellbores, and by applying the IMPES method elsewhere. The numerical formulation of the AIM has been discussed in detail by Thomas and Thurnau [110]. The first true AIM procedure was described by Forsythe and Sammon [48] for black oil simulation and by Collins *et al.* for compositional simulation [28, 128]. In 1993, Young and Russell implemented this method as an extension of the IMPES method to solve compositional problems [127, 128] in which they applied a strategy to minimize the overhead and accelerate the solution process. An IMPES-AIM model was developed by Cao and Aziz [15], which has been shown to

be more flexible, more stable, and much faster than the traditional AIM model. The stability of AIM will be addressed later in this dissertation. The biggest disadvantage of AIM is in the problem of the load balancing with parallel simulation. Since the implicitness and explicitness change dynamically, it is very hard to efficiently utilize all processors during simulation [7, 49, 57].

Recently developed reservoir simulators generally use mixed solution methodologies to solve reservoir problems. The combination of explicitness and implicitness sometimes produces very good results. However, these mixed concepts require the trade off of more complicated programing [15]. All of the solution methods discussed in this dissertation come with the finite difference method. Objectives about the applications of various finite element methods in reservoir simulation can be referred to the reference lists [4, 5, 8, 11, 30, 58, 88, 98, 102].

1.2.3 Iterative Coupling Technique

Iterative coupling is an operator splitting technique for solving multiphase flow systems. The primary procedure is to decouple the whole equation system into a pressure equation followed by saturation/concentration equations. These two stages are iteratively coupled at the end of each time step. Control of material balance error is the criterion to check for convergence of the current iteration. The necessity of an additional iteration is evaluated at the end of each iteration. The linear tolerance of the pressure solve is tightened gradually with each subsequent iteration. For a description of iterative coupling the reader may refer to Klie [71], Dawson, Klie, Wheeler and Woodward [31], and Lacroix et al. [75, 76].

In the late 1970's, Fussell proposed an iterative technique for compositional reservoir models [51]. In his work, he used an approach that minimized the number

of variables where simultaneous iteration was required. This method has shown quadratic convergence near the solution with the compositional model. Shenawi *et al.* used the iterative model to simulate imbibition phenomena in naturally fractured reservoirs [103]. This semi-analytical model was based on Buckley-Leverett flow in a fractured reservoir with water flooding. The iterative approach determined the water saturation distribution between the fracture and matrix in Laplace space.

We should note that the iterative coupling mentioned in this thesis is an numerical solution scheme like the FIM and the IMPES. Different from this, in recent years, much research work has focused on the iterative coupling of geomechanics with reservoir simulation for problems of reservoir compaction, surface subsidence, reservoir shear failure and reformation. These work can be referred to Gai [52], Gai, Dean, Wheeler and Lu [54], Tran, Nghiem and Buchanan [111, 112], Ji, Settari and Sullivan [65].

1.2.4 Linear Solver

The linear solver is the key factor that controls the performance of a reservoir model. The linear solver and its preconditioners have received particular attention since the earliest days of reservoir simulation. Several research efforts have addressed the development of solvers, including direct methods(Gaussian elimination) and iterative methods.

Brand and Heinemann presented an incomplete LU (ILU) factorization coupled with generalized conjugate-gradient (CG) acceleration to solve linear equations resulting from locally irregular grids in 1990. It was also useful in a regular grid. Compared to conventional ILU preconditionings, their method required very few iterations to reach convergence [12]. In the same year, Wong, Firoozabadi and Aziz

addressed the volume balance method with a comparison to the Newton-Raphson scheme [126]. The volume balance method was developed based on volume conservation, while the Newton method was developed based on mass conservation. However, an analytical demonstration proved both of the two methods lead to the same matrix system. At an earlier time, Watts used the preconditioned conjugate gradient(PCG) to solve the pressure equation in the IMPES model, and he made comparison with the strongly implicit procedure(SIP) [119]. According to his research, PCG was faster than SIP for 2D problem but slower than SIP for 3D problem when SIP worked well. Eisenstat, Elman and Schultz described a collection of block preconditioners for use in solving large, sparse linear systems [39]. This collection was compared with application of several point preconditioners. Eaton gave a detailed introduction of multigrid preconditioners for the two-phase flow problem in his dissertation in 2001 [37].

In the field of mathematics, Kwak conducted a research on the V-cycle multigrid for the cell-centred finite difference method (FD) [73, 74]. It was applied to second-order elliptic boundary value problems for both 2D and 3D. This approach converges quickly and it can be used as a good preconditioner.

The Center for Subsurface Modeling (CSM) at the University of Texas has done remarkable work on the development of solvers for reservoir simulation. Several solver packages have been utilized in the reservoir simulation tool, Integrated Parallel Accurate Reservoir Simulator (IPARS). Those applications can be referred to Lacroix, Vassilevski and Wheeler [75, 76], Vassilevski [115], and Kile and Wheeler [72]. Generalized Minimum Residual (GMRES) with various preconditioners on the pressure block, such as LSOR, multilevel ILU, AMG *et al.* is the most popular method in IPARS. Most of these solvers are designed for parallel simulation.

Killough and Wheeler discuss linear solver on parallel computation with domain decomposition algorithms [70].

Algebraic multigrid method (AMG) is known as the efficient solver and preconditioner for flow equation systems. The Fraunhofer SCAI develops a linear solver package with AMG for systems (SAMG), which has been successfully used in solving FIM and iterative systems [107, 99]. Parallel version of SAMG, SAMGp is also provided [100]. Details will be discussed later in the solver chapter. High Performance Preconditioners (HYPRE) is a library for solving large, sparse linear systems equations on parallel computers [62]. It includes several solvers and preconditioners with MPI parallisation. The application and performance of HYPRE will be discussed later in the solver chapter as well.

Fung and Al-Shaalan have proposed a parallel iterative solver for dual-porosity, dual-permeability reservoir simulation [50]. Their method involves the ordering of the system Jacobian matrix so that each column of cells is factorized exactly, and then the system is solved using the generalized conjugate residual (GCR) method.

1.2.5 Reservoir Simulation with HPC

With the increasingly accelerated development of multiple processor clusters, especially the distributed memory clusters, reservoir simulations on high performance computers have received more and more attention recently. Killough gave a brief discussion of the early attempts at HPC simulation [69], and pointed out three main concerns: load balance, data structure, and linear solvers. Since then, several reservoir simulators with parallel implementation have been developed and verified [21, 34, 47, 52, 81, 82, 93, 104, 129]. A parallel multiblock reservoir model for two-phase and black oil systems implemented on IPARS by Lu [78, 79] has shown high

performance with various numerical examples.

Magras, Quandalle and Bia have applied OpenMP instructions to ATHOS to form a parallel reservoir simulator [82]. OpenMP is designed for shared memory platform and it is easier to implement than message passing interface (MPI). However, clusters with a shared memory mode experience a scalability bottleneck when the number of processors reaches a certain level. Currently, research is ongoing to develop next generation reservoir simulators that are scalable on parallel computers. DeBaun *et al.* have designed an object-oriented, component-based architecture to utilize computational resources efficiently. Both static and dynamic load balancing are supported on structured and unstructed grids [34]. Shiralkar *et al.* have developed an unstructured simulator with parallel capability as well [104]. Their simulator integrates a surface network with the subsurface flow model to conduct a full-field simulation. Fjerstad *et al.* have run a parallel simulation with up to 64 processors using their next generation parallel simulators [47]. Parallel scalability becomes one of the common functionalities of new developed reservoir simulators.

Linear solvers for parallel reservoir simulation have been discussed in previous sections [31, 75, 81, 115].

1.2.6 IPARS Framework

All of the model developments and case studies that are discussed in this dissertation are dependent on the framework of IPARS, which is developed by Dr. John Wheeler and several other people from CSM. It is a next generation framework for developing and running reservoir simulations on both single processor and multiple processors environments. It has the fundamental functions of other commercial reservoir simulators, such as grid generation, MPI communication, memory allo-

cation, input/output control, visualization and table look-up *et al,* and it is also a good toolkit for reservoir simulation research, since the developer only needs to code the physical model for a specific application. Currently, there are several physical models in the IPARS package, including the air/water model, the oil/water model, the black oil and compositional models, the geomechanical model, as well as the chemical reactive transport model. Several other models are still under construction. IPARS uses Fortran 77 for most calculations of the individual models, C for memory management, MPI for parallel implementation, and C++ for mortar interface calculations.

IPARS has an operational history of more than 10 years and many people have contributed to this simulation toolkit in various aspects. The simulation performance of IPARS with various reservoir problems has been illustrated in several reports and papers [33, 64, 75, 77, 79, 80, 90, 92, 115, 123]. The reader is encouraged to refer to the references for further details.

1.3 Dissertation Outline

After the introduction, this dissertation is composed of five additional chapters, and it is organised as follows:

1. In Chapter 2, the iterative models for two-phase flow including oil/water and air/water flow are developed and implemented. The advanced numerical techniques that are applied in the iterative model are addressed in detail. The stability and efficiency of the iterative model are also discussed. Numerical experiments are conducted to verify these advantages.

2. Chapter 3 presents the development process of the iterative black oil model. Numerical examples using relevant problem statements are included.

3. In Chapter 4, linear solvers and preconditioners that have been applied to the iterative model are discussed and compared.

4. In Chapter 5, parallel implementation issues for the iterative model are addressed. In particular, analysis of the parallel performance of the iterative model is conducted, based on numerical examples with thousands of processors and millions of elements.

5. Chapter 6 concludes the dissertation with a summary of accomplishments and recommendations for future work.

Chapter 2

Iterative Two-phase Model

This chapter will discuss and explain the concepts of iterative coupling with applications to two-phase flow problems. Both the FIM and the IMPES method are discussed and compared to verify the performance of the iterative model. Development of the new models is based on the oil/water IMPES model and the air/water FIM model in the IPARS framework, respectively [64, 90, 123].

2.1 Assumptions

Two-phase flow in this dissertation refers to oil/water and air/water flows. All of these phases are immiscible and slightly compressible. The reservoir is isothermal, with constant pore volume during production. Both gravity and capillary pressure exist.

2.2 Iterative Oil/Water Model

2.2.1 Numerical Formulation

Pressure Equation

The basic governing equations of the two-phase oil/water model are the mass conservation equation and the transport equation, also known as the Darcy flow equation. For compressible flow, we have

$$\frac{\partial(\phi\rho_w S_w)}{\partial t} = - \bigtriangledown \cdot(\rho_w u_w) + q_w \tag{2.1}$$

for the water phase and

$$\frac{\partial(\phi\rho_o S_o)}{\partial t} = - \bigtriangledown \cdot(\rho_o u_o) + q_o \tag{2.2}$$

for the oil phase.

Darcy's law for fluid flow in porous media is displayed as

$$u_\alpha = -K\lambda_\alpha(\nabla P_\alpha - \rho_\alpha g \nabla Z), \tag{2.3}$$

where α represents the water and oil phase, λ is the fluid mobility, and u_α is the flow volumetric velocity.

The capillary pressure relationship and saturation constraint are given by,

$$P_{cow} = P_o - P_w \tag{2.4}$$

$$S_w + S_o = 1 \tag{2.5}$$

Summing equations (2.1) and (2.2) together, and then substituting Darcy's law into them, we obtain

$$\frac{\partial\left[\phi\left(\rho_w S_w+\rho_o S_o\right)\right]}{\partial t}=\bigtriangledown\cdot\left[K\lambda_w\rho_w(\nabla P_w-\rho_w g\nabla Z)\right] + \bigtriangledown\cdot\left[K\lambda_o\rho_o(\nabla P_o-\rho_o g\nabla Z)\right]+q_w+q_o. \tag{2.6}$$

Substituting the capillary pressure Eq.(2.4) into Eq.(2.6), then the right-hand-side of the above equation may be rewritten as

$$Rhs=\bigtriangledown\cdot\left[K(\lambda_w\rho_w+\lambda_o\rho_o)\nabla P_w\right]+\bigtriangledown\cdot\left[K\rho_o\lambda_o\nabla P_{cow}\right] - \bigtriangledown\cdot\left[K(\lambda_w\rho_w^2+\lambda_o\rho_o^2)g\nabla Z\right]+q_w+q_o. \tag{2.7}$$

Considering that density is a function of pressure and assuming the reference pressures for oil and water are under standard condition, we have

$$\rho_w=\rho_{w,ref}(1+C_w P_w) \tag{2.8}$$

$$\rho_o=\rho_{o,ref}(1+C_o P_o)=\rho_{o,ref}[1+C_o(P_w+P_{cow})]. \tag{2.9}$$

The accumulation term in Eq.(2.6) can be represented as

$$\rho_w S_w+\rho_o S_o=\rho_{w,ref}S_w(1+C_w P_w)+\rho_{o,ref}S_o(1+C_o P_w)+\rho_{o,ref}S_o C_o P_{cow}. \tag{2.10}$$

The discrete formula of the left-hand-side of Eq.(2.6) will be

$$Lhs=\frac{\left[\phi\left(\rho_w S_w+\rho_o S_o\right)^{k+1}\right]-\left[\phi\left(\rho_w S_w+\rho_o S_o\right)^{n}\right]}{\triangle t^{n+1}}$$

$$= \frac{\phi\left(\rho_{w,ref}S_wC_w + \rho_{o,ref}S_oC_o\right)^k P_w^{n+1}}{\triangle t^{n+1}} + \frac{\phi\rho_{o,ref}S_o^kC_oP_{cow}^k}{\triangle t^{n+1}}$$
$$+ \frac{\left[\phi\left(\rho_{w,ref}S_w + \rho_{o,ref}S_o\right)^k\right] - \left[\phi\left(\rho_wS_w + \rho_oS_o\right)^n\right]}{\triangle t^{n+1}}. \quad (2.11)$$

Collecting both the right- and left-hand-side terms that involve water pressure into the coefficient matrix, and moving all other terms to the right-hand-side, we obtain a discretized pressure equation as follows:

$$\begin{aligned} -\frac{\phi\left(\rho_{w,ref}C_wS_w^k + \rho_{o,ref}C_oS_o^k\right)P_w^{n+1}}{\triangle t^{n+1}} &+ \bigtriangledown\cdot\left[K\left(\rho_w^k\lambda_w^k + \rho_o^k\lambda_o^k\right)\nabla P_w^{n+1}\right] + q_w^{k+1} + q_o^{k+1} \\ = \frac{\phi\left(\rho_{w,ref}S_w^k + \rho_{o,ref}S_o^k\right)}{\triangle t^{n+1}} &+ \frac{\phi\rho_{o,ref}S_o^kC_oP_{cow}^k}{\triangle t^{n+1}} - \frac{\phi\left(\rho_wS_w + \rho_oS_o\right)^n}{\triangle t^{n+1}} \\ -\bigtriangledown\cdot\left(K\rho_o^k\lambda_o^k\nabla P_{cow}\right) &+ \bigtriangledown\cdot\left[K\left(\left(\rho_w^2\lambda_w\right)^k + \left(\rho_o^2\lambda_o\right)^k\right)g\nabla Z\right]. \quad (2.12) \end{aligned}$$

The well term for the water phase can be discretized as follows:

$$q_w^{n+1} = \frac{GLKk_{rw}}{\mu_w}\left(P_{wb} - P_w^{k+1} - \rho_wg\nabla Z\right)$$
$$= -\frac{GLKk_{rw}}{\mu_w}P_w^{k+1} + \frac{GLKk_{rw}}{\mu_w}\left(P_{wb}^k - \rho_w^kg\nabla Z\right). \quad (2.13)$$

Similarly, the well term for the oil phase is given by:

$$q_o^{n+1} = \frac{GLKk_{ro}}{\mu_o}\left(P_{wb} - P_w^{k+1} - P_{cow}^k - \rho_og\nabla Z\right)$$
$$= -\frac{GLKk_{ro}}{\mu_o}P_w^{k+1} + \frac{GLKk_{ro}}{\mu_o}\left(P_{wb}^k - P_{cow}^k - \rho_o^kg\nabla Z\right). \quad (2.14)$$

Saturation Equation

To simplify the formulation of the saturation equation, here we define some variables to denote mass velocities U_o, U_w and U_t, where $U_w = \rho_w u_w$, $U_o = \rho_o u_o$ and $U_t = U_o + U_w$.

We also define the mass mobility as $\Lambda_o = \lambda_o \rho_o$, $\Lambda_w = \lambda_w \rho_w$, and the total mass mobility as $\Lambda_t = \Lambda_w + \Lambda_o$.

We expand the total mass velocity as following:

$$\begin{aligned} U_t = U_w + U_o = & - K(\Lambda_w + \Lambda_o)\nabla P_w - K\Lambda_o \nabla P_{cow} \\ & + K(\Lambda_w \rho_w + \Lambda_o \rho_o) g \nabla Z. \end{aligned} \tag{2.15}$$

The water mass velocity is given by

$$\begin{aligned} U_w = \rho_w u_w &= \rho_w \left[-K\lambda_w \left(\nabla P_w - \rho_w g \nabla Z \right) \right] \\ &= -K\Lambda_w \nabla P_w + K\Lambda_w \rho_w g \nabla Z. \end{aligned} \tag{2.16}$$

Substituting Eq.(2.16) into the total mass velocity Eq.(2.15) to obtain

$$\begin{aligned} U_t = \frac{\Lambda_t}{\Lambda_w} U_w - K\Lambda_t \rho_w g \nabla Z - K\Lambda_o \nabla P_{cow} \\ + K\left(\rho_w \Lambda_w + \rho_o \Lambda_o\right) g \nabla Z. \end{aligned} \tag{2.17}$$

Thus Eq.(2.16) becomes

$$U_w = \frac{\Lambda_w}{\Lambda_t} U_t + K \frac{\Lambda_w \Lambda_o}{\Lambda_t} \left[\nabla P_{cow} + \left(\rho_w + \rho_o\right) g \nabla Z \right]. \tag{2.18}$$

Substituting U_w into the water phase mass conservation Eq. (2.1), we obtain the saturation equation:

$$\frac{\partial\phi\left(\rho_w S_w\right)}{\partial t}+\bigtriangledown\cdot\left[K\frac{\Lambda_w\Lambda_o}{\Lambda_t}\nabla P_{cow}\right]+\nabla\cdot\left(\frac{\Lambda_w}{\Lambda_t}U_t\right)$$
$$+\bigtriangledown\cdot\left[K\frac{\Lambda_w\Lambda_o}{\Lambda_t}\left(\rho_w-\rho_o\right)g\nabla Z\right]=q_w. \tag{2.19}$$

Here the discretized form is written as:

$$\frac{\phi\rho_w^{n+1}S_w^{n+1}}{\triangle t^{n+1}}+\bigtriangledown\cdot\left[K\left(\frac{\Lambda_o\Lambda_w}{\Lambda_t}\right)^{n+1}\left(\frac{dP_{cow}}{dS_w}\right)^{n}\nabla S_w^{n+1}\right]-q_w^{n+1}$$
$$=\frac{\phi\rho_w^{n}S_w^{n}}{\triangle t^{n+1}}-\nabla\cdot\left(\frac{\Lambda_w}{\Lambda_t}U_t\right)^{n+1}-\bigtriangledown\cdot\left[K\left(\frac{\Lambda_o\Lambda_w}{\Lambda_t}\right)^{n+1}\left(\rho_w^{n+1}-\rho_o^{n+1}\right)g\nabla Z\right]. \tag{2.20}$$

Eqs.(2.12) and (2.20) are the formulations used in the iterative oil/water model.

2.2.2 Algorithms

In Figure 2.1 we describe a typical time step in the iterative oil/water model. In each time step, the pressure equation is solved implicitly, followed by the saturation equation, which is solved semi-implicitly. These two stages are iteratively coupled at the end of the time step. Additional iterations are conducted till the convergence criterion of material balance errors has been satisfied. The simulation procedure and most of the subroutines within one time step are listed in Appendix I.

2.3 Iterative Air/Water Model

The governing equations of the iterative air/water model are very similar to those of the oil/water model in Eq.(2.1-2.3). The only difference is in the expression of

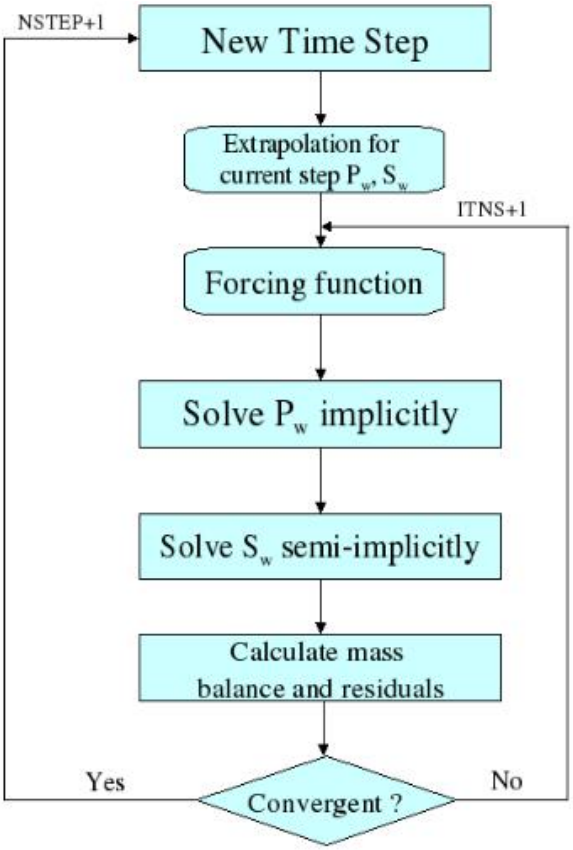

Figure 2.1: Flow chart of the iterative oil/water model

air phase density, which is nonlinear rather than linear

$$\rho_a = \frac{P_a M}{Z(P_a)RT}, \tag{2.21}$$

where ρ_a is the air phase density, P_a is the air phase pressure, M is the molecular weight of air, R is the air constant, T is the temperature, and $Z(P_a)$ is the compressibility factor. Again, P_w and S_w are chosen as the primary variables.

Recall that the mass conservation equation and the Darcy equation for two-phase flow are as follows:

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \nabla\cdot(\rho_\alpha u_\alpha) = \rho_\alpha q_\alpha, \quad \alpha = a, w \tag{2.22}$$

$$u_\alpha = -\rho_\alpha K\frac{k_{r\alpha}}{\mu_\alpha}(\nabla P_\alpha - \rho_\alpha g\nabla Z), \quad \alpha = a, w. \tag{2.23}$$

For the water equation, substituting the linear density into the mass conservation equation produces

$$\begin{aligned}\frac{\partial[\phi\rho_{w,ref}(1+C_wP_w)S_w]}{\partial t} &- \nabla\cdot[\rho_{w,ref}(1+C_wP_w)K\frac{k_{rw}}{\mu_w}(\nabla P_w - \rho_w g\nabla Z)] \\ &= \rho_{w,ref}(1+C_wP_w)q_w.\end{aligned} \tag{2.24}$$

For the air equation, substituting the nonlinear density and the capillary pressure into the mass conservation equation produces

$$\begin{aligned}\frac{\partial[\phi\frac{P_aM}{Z(P_a)RT}S_a]}{\partial t} &- \nabla\cdot[K\frac{P_aM}{Z(P_a)RT}\frac{k_{ra}}{\mu_a}(\nabla P_w + \nabla P_{caw} - \rho_a g\nabla Z)] \\ &= \frac{P_aM}{Z(P_a)RT}q_a.\end{aligned} \tag{2.25}$$

Since the water phase and the air phase have very different viscosity and density

values, if Eq.(2.24) and Eq.(2.25) are summed together as in the oil/water model, several features of the air phase can be neglected, which will reduce the accuracy of the solutions. In order to avoid this imbalance problem, phase scaling is required.

Multiplying Eq.(2.24) by $\frac{\mu_w}{\rho_{w,ref}}$ and discretizing by time, we get

$$\begin{aligned}
\phi\mu_w C_w S_w^{n+1,k} P_w^{n+1,k+1} &- \triangle t^{n+1}\nabla\cdot[K(1+C_w P_w^{n+1,k})k_{rw}^{n+1,k}]\nabla P_w^{n+1,k+1} \\
&- \triangle t^{n+1}\mu_w(1+C_w P_w^{n+1,k})q_w^{n+1,k+1} \\
&= -\triangle t^{n+1}\nabla\cdot[K(1+C_w P_w^{n+1,k})k_{rw}^{n+1,k}\rho_w^{n+1,k}g\nabla Z] \\
&+ \frac{\mu_w}{\rho_{w,ref}}(\phi\rho_w S_w)^n - \phi\mu_w S_w^{n+1,k}, \qquad (2.26)
\end{aligned}$$

where $n+1$ is the current time step and k and $k+1$ are the iteration numbers within time step $n+1$.

Multiplying Eq.(2.25) by $\frac{\mu_a}{\rho_a^{n+1,k}}$ and discretizing by time, we get

$$\begin{aligned}
\phi\mu_a S_a^{n+1,k}\frac{P_w^{n+1,k+1}}{P_a^{n+1,k}} &- \triangle t^{n+1}\nabla\cdot[Kk_{ra}^{n+1,k}\nabla P_w^{n+1,k+1}] \\
&- \triangle t^{n+1}\mu_a q_a^{n+1,k+1} \\
&= \triangle t^{n+1}\nabla\cdot[Kk_{ra}^{n+1,k}(\nabla P_{caw}^{n+1,k} - \rho_a^{n+1,k}g\nabla Z)] \\
&+ \frac{\mu_a}{\rho_a^{n+1,k}}(\phi\rho_a S_a)^n - \phi\mu_a S_a^{n+1,k}\frac{1}{P_a^{n+1,k}}P_{caw}^{n+1,k}. \qquad (2.27)
\end{aligned}$$

Summing the above two equations together, and collecting terms involving P_w, we obtain the pressure equation for the air/water model given by

$$\begin{aligned}
&- [\phi\mu_w C_w S_w^{n+1,k} + \phi\mu_a S_a^{n+1,k}\frac{1}{P_a^{n+1,k}}] \times P_w^{n+1,k+1} \\
&+ \triangle t^{n+1}\nabla\cdot[K((1+C_w P_w^{n+1,k})k_{rw}^{n+1,k} + k_{ra}^{n+1,k})\nabla P_w^{n+1,k+1}] \\
&+ \triangle t^{n+1}[\mu_w(1+C_w P_w^{n+1,k})q_w^{n+1,k+1} + \mu_a q_a^{n+1,k+1}]
\end{aligned}$$

$$
\begin{aligned}
&= -\frac{\mu_w}{\rho_{w,ref}}(\phi\rho_w S_w)^n + \phi\mu_w S_w^{n+1,k} - \frac{\mu_a}{\rho_a^{n+1,k}}(\phi\rho_a S_a)^n + \phi\mu_a S_a^{n+1,k}\frac{1}{P_a^{n+1,k}}P_{caw}^{n+1,k} \\
&+ \triangle t^{n+1}\nabla\cdot[K(1+C_w P_w^{n+1,k})k_{rw}^{n+1,k}\rho_w^{n+1,k}g\nabla Z] \\
&- \triangle t^{n+1}\nabla\cdot[Kk_{ra}^{n+1,k}(\nabla P_{caw}^{n+1,k} - \rho_a^{n+1,k}g\nabla Z)]. \qquad (2.28)
\end{aligned}
$$

The saturation equation of the air/water model is the same as that of the oil/water model in Eq.(2.20).

According to our experience, phase scaling affects the iterative model significantly. It speeds up the simulation convergence and reduces the number of iterations within the given time step. Phase scaling will be extended to the black oil model later in this dissertation.

2.4 Numerical Techniques

Three numerical techniques have been applied to the iterative model, and they have been proven to be effective in enhancing simulation performance. These techniques, extrapolation, forcing function and Kirchhoff transformation have been incorporated into the iterative procedure.

2.4.1 Extrapolation

In the conventional IMPES two-phase model, fluids are treated as incompressible or slightly compressible, and certain densities are divided from both sides of the mass conservation equations before they are summed together. The pressure equation may then be solved easily with saturation and fluid properties upwinded from previous time steps. However, this method provides large inaccuracies in deep reservoirs

with very high pore pressures and highly compressible fluids, as well as when fluid properties are changing rapidly.

In the iterative two-phase model, the saturations and densities are involved in the pressure solve in Eq.(2.12). In this way, both saturation and density must be initially estimated and explicitly used in the pressure equation based on the previous time steps. The initial guess must be as close as possible to the exact reservoir properties. In order to improve the accuracy of the initial estimation, the initial estimations of pressure and saturation are obtained by extrapolating the pressure and saturation values from the previous two time steps. The resulting values are more accurate than simply upwinding the values from the last time step. Here we set

$$S_w^{n+1} = \alpha S_w^n + \beta S_w^{n-1}, \tag{2.29}$$

$$P_w^{n+1} = \alpha P_w^n + \beta P_w^{n-1}, \tag{2.30}$$

where

$$\alpha = \frac{\triangle t^n + \triangle t^{n-1}}{\triangle t^{n-1}},\ \beta = -\frac{\triangle t^n}{\triangle t^{n-1}}, \tag{2.31}$$

and $\triangle t^n$ is the size of time step n.

Thus at each time step of the iterative model, initial values of P_w^{n+1} and S_w^{n+1} are first obtained based on extrapolation; density ρ^{n+1} and mobility λ^{n+1} are then updated sequentially using these values, both of which will be applied in the pressure solve.

2.4.2 Forcing Function

Both in the iterative model and in the IMPES model, the pressure equation is solved implicitly, and it takes much more CPU time than the saturation equation [78]. However, the pressure at the beginning of each time step cannot be solved exactly accurate because of the inaccurate upwinded or extrapolated initial values. In order to save CPU time and make the iterative method more efficient, we apply a forcing function to adaptively change the linear tolerance for the pressure solve [31, 71, 76], using a loose tolerance at the beginning and then tightening it gradually in following iterations until the convergence criteria has been satisfied. The ease of a forcing function for solving nonlinear equations was first formulated by Eisenstat and Walker [40]; the effectiveness of this approach was demonstrated by Dawson *et al.*, and Vassilevski et al. for two and three phase FIM formulations respectively [31, 75, 115]. The choice of the forcing term is as follows:

$$\eta^{(k)} = max\left\{\eta^{*(k)}, \gamma\left(\eta^{(k-1)}\right)^2\right\}, \tag{2.32}$$

where

$$\eta^{*(k)} = \gamma\left(\frac{\left\|Resid^{(k)}\right\|}{\left\|Resid^{(k-1)}\right\|}\right)^2, \tag{2.33}$$

Here γ is a constant close to 1 and typically taken to be 0.9. *Resid* is the computation residual for each pressure solve (PCG or GMRES iteration). So with the forcing term in the pressure solve, we first give an initial linear tolerance $PWTOL^{(1)}$ based on the required tolerance in the simulation, and tighten it with the following formula using the increasing iteration number k

$$PWTOL^{(k)} = \eta^{(k)} * PWTOL^{(k-1)}. \tag{2.34}$$

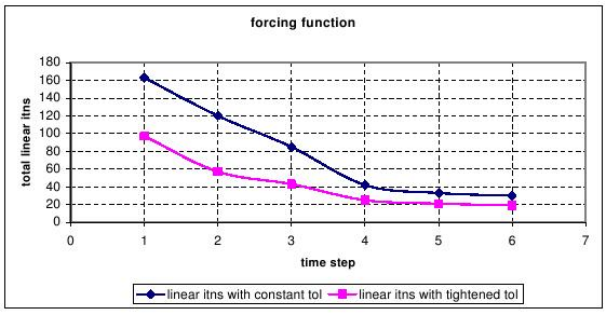

Figure 2.2: Forcing function in the iterative model

The application of a forcing function in the pressure solve shortens the simulation time remarkably by cutting down on the number of iterations per time step. Figure 2.2 shows the total iterations of each time step for a given simple benchmark problem. For time step 1, the total linear iterations without a forcing function is 160. The application of forcing function decreases this number to 100. All the following time steps also show the average number of linear iterations for a model with a forcing function is far less than that of a model without forcing function.

2.4.3 Kirchhoff Transformation

Considering the saturation equation that we derived for the iterative two-phase model in Eq.(2.20), the derivative of capillary pressure with respect to the saturation may exhibit degenerate diffusion, which causes obvious numerical oscillation. We could apply the Kirchhoff transformation to avoid this problem [37, 42] and also provide an additional approximation option beside upwinding. The Kirchhoff transformation substitutes the capillary pressure term in the saturation equation

with the diffusion function $D(S_w)$.

$$D(S_w) = -\int_0^{S_w} \alpha(\xi)\frac{\partial P_c(\xi)}{\partial \xi}d\xi, \tag{2.35}$$

where

$$\begin{aligned} \alpha(S_w) &= \beta(S_w)\Lambda_o(S_w), \\ \beta(S_w) &= \frac{\Lambda_w}{\Lambda_t}. \end{aligned} \tag{2.36}$$

This function D is S-shaped and smooth for all values of $S_w \in (S_{wi}, 1 - S_{or})$.

Substituting this function into the saturation equation of the iterative model, Eq.(2.20), we obtain that:

$$\begin{aligned} \frac{\phi\rho_w^{n+1}S_w^{n+1}}{\triangle t^{n+1}} &- \bigtriangledown \cdot [K\nabla \cdot D(S_w)] - q_w^{n+1} = \frac{\phi\rho_w^n S_w^n}{\triangle t^{n+1}} \\ &- \nabla \cdot \left(\frac{\Lambda_w}{\Lambda_t}U_t\right)^n - \nabla \cdot \left[K\left(\frac{\Lambda_o\Lambda_w}{\Lambda_t}\right)^n (\rho_w^n - \rho_o^n)\, g\nabla Z\right]. \end{aligned} \tag{2.37}$$

If an effective mass flux is defined as

$$\psi = \left[\beta(S_w)U_t - K\nabla \cdot D(S_w) + \beta(S_w)\Lambda_o\left(\rho_w - \rho_o\right)Kg\nabla Z\right], \tag{2.38}$$

then the saturation equation can be rewritten explicitly as

$$\rho_w^{n+1}S_w^{n+1} = \rho_w^n S_w^n - \frac{\triangle t^{n+1}}{\phi}\nabla \cdot \psi + \frac{\triangle t^{n+1}}{\phi}q_w^{n+1}. \tag{2.39}$$

A data set of the Kirchhoff transformation is obtained before the simulation. The existing data is used to perform a five-point Gauss-quadrature rule approximation

to the definite integral at a discrete set of points. Cubic spline interpolation is then applied to this set during the course of simulation to produce the necessary values. IPARS provides a table-lookup infrastructure that perfectly matches the requirements of this approach [37, 77]. A table of function values over the attainable values of $S_w \in (S_{wi}, 1 - S_{or})$ is calculated and stored in an array during initialization; the program searches the corresponding values during simulation for the given saturations. No calculation of the Kirchhoff transformation occurs once simulation starts.

The Kirchhoff transformation is currently applied in the iterative oil/water model. Improvements in model stability have been obtained with the application of Kirchhoff transformation.

2.5 Stability Study

The stability of a numerical method is related to the growth, decay and accumulation of perturbations over many time steps. It is determined by the properties of the differential equation and the discretization scheme [41]. For example, parabolic problems may allow long time approximation of solutions by some particular discretization methods. One numerical scheme is stable if errors introduced in each time step do not amplify during subsequent computation. Stability in a reservoir simulation is reflected to the degree that the iteration convergence speed is insensitive to time step size. Stability of a nonlinear system is the necessary condition for obtaining reasonable solutions. An unstable scheme yields results that are meaningless.

2.5.1 FIM

The FIM scheme is unconditionally stable if all the terms are treated implicitly, and there are no stability restrictions on the simulation. However, very large time steps may either introduce discretization errors or require too many iterations to solve the nonlinear equations at a given time. Generally, there are two time step controls that are used in all implicit scheme:

- Limit on maximum change of parameters;
- Limit on time truncation error.

The limitation of change in parameters is discussed by Aziz and Settari [6] and Coats et al. [25, 26]. Wan *et al.* proposed a general stability criterion for the black oil model with various degrees of implicitness [116], in which a Fourier stability analysis was used for explicit schemes. Aziz and Settari derived the criteria to control numerical behavior to produce a stable solution, with which the new time step had the form:

$$\triangle t^{n+1} = \triangle t^n \left[\frac{(1+\omega)\,\eta_i}{\delta_i + \omega\eta_i} \right]_{min\ over\ i}, \tag{2.40}$$

where i is the block number, η_i is the desired change, δ_i is the change over $\triangle t^n$, and ω is a constant between 0 and 1. η_i usually is the same for all blocks but different for different variables. For pressure and saturation, the typical values of η for pressure and saturation respectively were:

- $\eta_p = 50$ to $500psi$,
- $\eta_s = 0.05$ to 0.5.

In IPARS, the FIM models for two-phase and black oil flow are well developed with dynamic time step control functionality, using material balance to control convergence, instead of using the above time step control method. In numerical example studies in this dissertation, the solution found using the FIM model is treated as the true solution, and it is then used to verify the accuracy of the iterative model.

2.5.2 IMPES Model

The instability of the IMPES method comes from the explicit formula of capillary pressure and transmissibility[6]. For 1-D incompressible multiphase flow, the time step limitation due to the explicitness of capillary pressure is

$$\triangle t < \frac{4V\phi}{\gamma_1 \left|P'_{cow}\right| T_w \left(\frac{T_o+T_q}{T_T}\right) + \gamma_3 \left|P'_{cog}\right| T_g \left(\frac{T_o+T_w}{T_T}\right) + \frac{1}{T_T}\sqrt{X}}, \tag{2.41}$$

where

$$\begin{aligned} X = \gamma_1 \left|P'_{cow}\right| T_w(T_o+T_g) &- \gamma_3 \left|P'_{cog}\right| T_g(T_o+T_w)^2 + 4\left|P'_{cow}\right|\left|P'_{cog}\right| T_w^2 T_g^2 \gamma_1\gamma_2 \\ \gamma_m &= 4\sin^2\left(\frac{\alpha_m}{2}\right), \\ \alpha_m &= m\triangle x_i, m = 1,2,...,n_x, \end{aligned}$$

in which T_α is the transmissbility of phase α.

For a two-phase system, the stability with respect to explicitness of transmissbility leads to the condition

$$\triangle t < \frac{V\phi}{\left(\frac{df_w}{dS_w}\right) q_T}. \tag{2.42}$$

Dividing by $A\phi$, we have

$$\triangle t \frac{df_w}{dS_w} \frac{u_T}{\phi} < \triangle x, \tag{2.43}$$

where

$$\frac{df_w}{dS_w} \frac{u_T}{\phi} = \left(\frac{\partial x}{\partial t}\right)_{S_w = const} = u_s$$

is the velocity of the water front with constant saturation.

This is the main restriction of the IMPES method, and it also applies to multi-phase multi-dimensional flow. The time step size $\triangle t$ is tightly limited by the above equations. The IMPES method has been proven to have a significant stability problem when dealing with complicated reservoir problems [15, 25, 26].

2.5.3 Iterative Model

In the iterative model, the capillary pressure is treated implicitly, as in the saturation Eq.(2.20). In this case, the simulation is more stable than IMPES. At the same time, the forcing function allows for smooth parameter changes in sequential iterations. In this dissertation, a $CFL - type$ stability control strategy is set for the purpose of comparing it with the IMPES method. We set

$$CFL = \frac{u_t}{\phi} \frac{\triangle t}{\triangle x}, \tag{2.44}$$

where u_t is the total velocity, and $\triangle x$ is the geometric dimension in the direction of flow.

2.6 Numerical Experiments

Several numerical experiments have been run with the iterative method, with the fully implicit method, and with the IMPES method. All of these models are developed in the IPARS framework.The FIM has been well tested and shows very close solutions with other commercial reservoir simulators [52, 64, 77, 78]. In this dissertation, the FIM is used as the standard solution to verify the performance of the iterative model. Eclipse is also used in the comparison study. Both accuracy and efficiency are considered during the comparisons, in terms of material balance errors and CPU time used.

The 10th SPE comparative project (SPE 10) describes a large reservoir with $1,122,000$ grid elements with high heterogeneity[22]. Some of the numerical examples designed in this dissertation are based on this reservoir, using both the original and an upscaled mesh. The upscaling of porosity is based on a mathematical averaging method. The upscaling of permeability uses both mathematical and harmonic averaging methods. If the upscaling direction is the same as the permeability direction, harmonic averaging is used; otherwise mathematical averaging is used [91].

In the case of calculating the average permeability of grid blocks in series, harmonic averaging is adopted as follows:

$$\overline{k} = \frac{L}{\frac{L_1}{k_1} + \frac{L_2}{k_2} + \cdots + \frac{L_n}{k_n}}. \tag{2.45}$$

In the case of calculating the average permeability of grid blocks in parallel, mathematical averaging is adopted as follows:

$$\overline{k} = \frac{L_1}{L}k_1 + \frac{L_2}{L}k_2 + \cdots + \frac{L_n}{L}k_n, \tag{2.46}$$

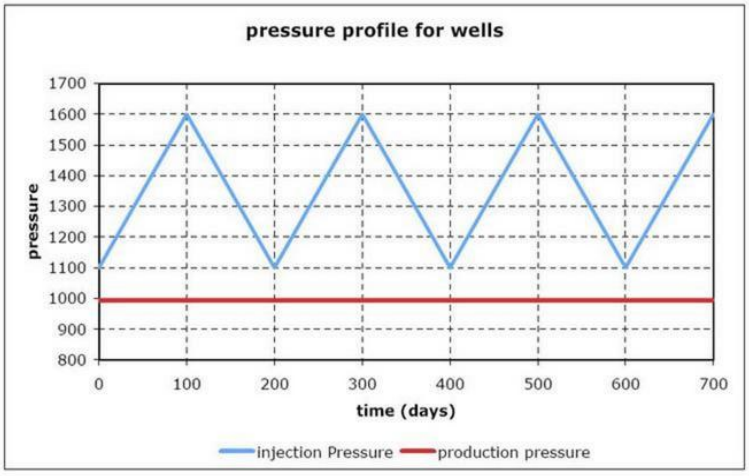

Figure 2.3: Well bottom hole pressure

where $\overline{k}$ is the average permeability, $k_{i=1,2,...n}$ is the permeability for each of the grid blocks, and $L_{i=1,2,...n}$ is the length of each of the grid blocks.

2.6.1 Example 1: 2D Problem

This example is a $2D$ two-phase oil/water problem with heterogeneous permeability field $L_x \times L_y$. The total number of mesh elements is 2000, with $L_x = 100$ and $L_y = 20$. P_o and S_w are specified at $x = 0$ and $x = L_x$. One injector is located at $x = 0, y = 10$ and one producer at $x = 100, y = 10$. Both wells are bottom hole pressure specified. The bottom hole pressure at the injector changes periodically from $1100psi$ to $1600psi$ while the pressure at the producer is kept constant at $1000psi$ as shown in Figure 2.3. FIM, IMPES and the iterative models are used to simulate this reservoir for 100 days using various time step controls.

The changing bottom hole pressure leads to fluctuations in the production rate. The iterative method produces a result very similar to that of the FIM, in

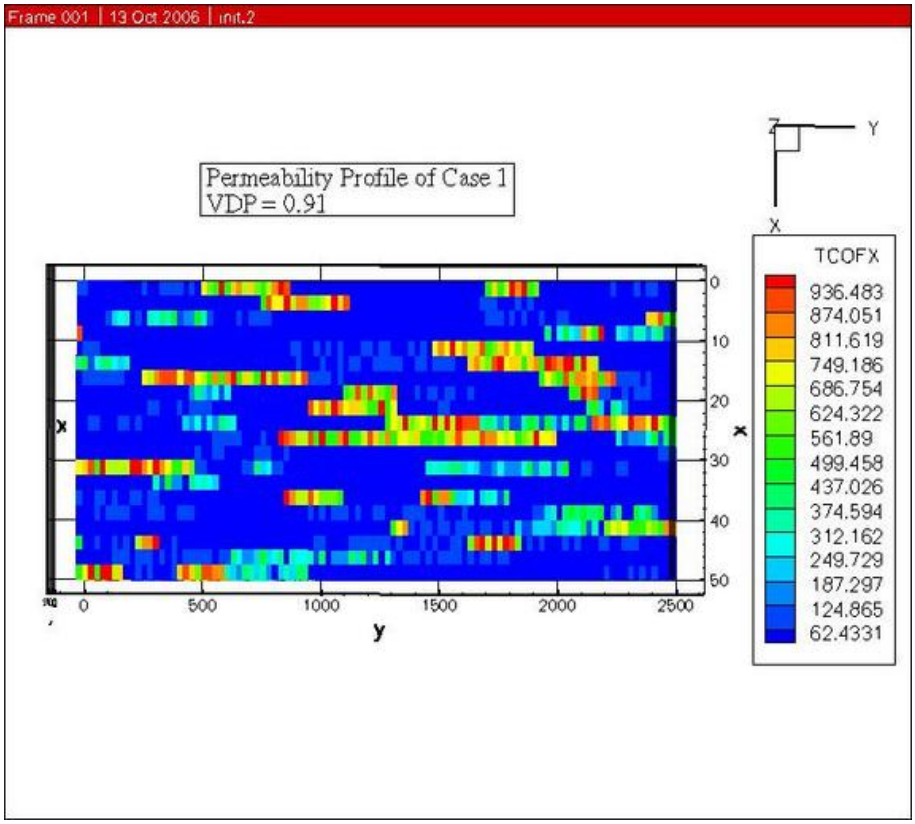

Figure 2.4: Reservoir permeability for example 1

Max time step	Model	Oil balance	Water balance	CPU time (sec.)
0.1 days	IMPES	–	–	173.319
2 days	Iterative	1.0000028	1.0000083	164.269
	FIM	1.00000267	1.0000011	685.505
5 days	Iterative	1.00000144	0.99999884	190.270
	FIM	1.00000133	0.99999855	769.424
10 days	Iterative	1.00000075	0.99999665	213.505
	FIM	1.00000102	0.99999719	720.190

Table 2.1: Performance comparison for example 1

Figure 2.5. The IMPES model is too unstable to complete the simulation with a reasonable solution (Figure 2.6). Table 2.1 gives a comparison of timings for the FIM and the iterative method with maximum time steps of 2 days, 5 days and 10 days, respectively.

Generally, the iterative method runs about $3 \sim 4$ times faster than the FIM. In addition, the iterative method achieves excellent material balance, with errors less than 10^{-6}. When the time step becomes larger, the ratio of CPU time taken by the FIM to that of the iterative method becomes smaller, which implies that the iterative method is more sensitive to time step size than the FIM. Thus the selection of a maximum time step size is important to enable the iterative method to achieve optimal performance.

2.6.2 Example 2: Upscaled SPE 10

This reservoir is taken from the SPE 10 comparative project; the upper part of the whole reservoir is used in this example [22]. The reservoir size is $2200ft \times 1200ft \times 100ft$, and the element dimension is $40ft \times 40ft \times 20ft$, total number of elements is 8250. The input data for relative permeabilities is listed in Table 2.2,

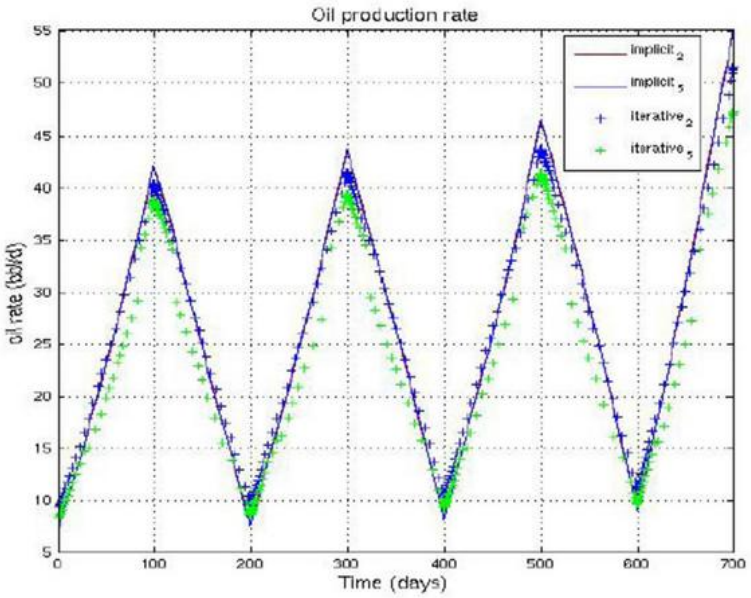

Figure 2.5: Oil rate in producer

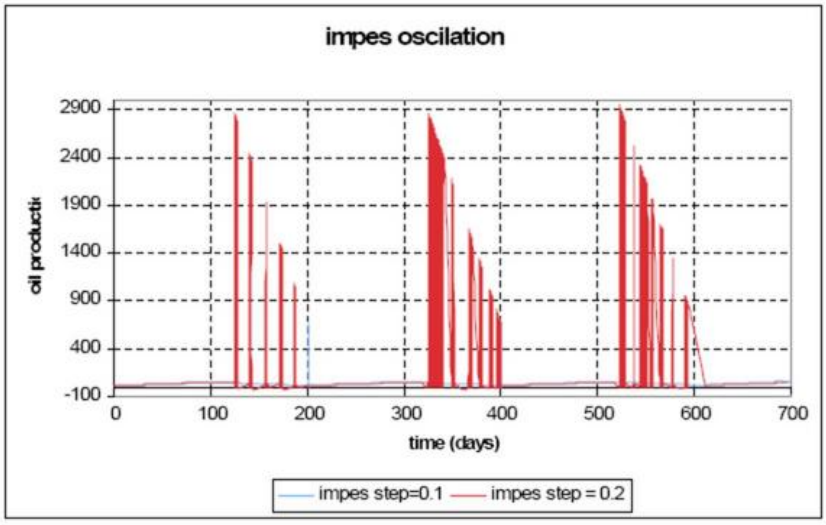

Figure 2.6: Oscillation of the IMPES method

S_w	k_{rw}	k_{ro}
0.20	0.0	1.0
0.25	0.0069	0.8403
0.30	0.0278	0.6944
0.35	0.0625	0.5625
0.40	0.1111	0.4444
0.45	0.1736	0.3403
0.50	0.2500	0.2500
0.55	0.3403	0.1736
0.60	0.4444	0.1111
0.65	0.5625	0.0625
0.70	0.6944	0.0278
0.75	0.8403	0.0069
0.80	1.0	0.0

Table 2.2: Relative permeabilities of SPE 10

and the capillary pressure curve is shown in Figure 2.7. For those points with water saturation values beyond this table, extrapolation with constant value has been applied to avoid the possible simulation failure. One water injection well is located at the center of the reservoir; four producers are located at the four corners of the reservoir. All of these wells are bottom hole pressure specified. The total simulation time is 2000 days and the reservoir properties are modified based on the original description given in SPE 10. All of the three solution methods are computed for this problem used in simulation with various time step controls. The timing comparisons are shown in Table 2.3. The IMPES method was unable to complete the simulation due to an unexpected oscillation.

The iterative model achieves solutions that are very close to the solutions given by the FIM. However, it just takes about half of the simulation time as compared to FIM when we choose time step as 2 days. After 2000 days of simulation, the iterative method keeps stable and its solution reflects the heterogeneity of the reservoir as

Max time step	Model	Oil balance	Water balance	CPU time (sec.)
2 days	Iterative	0.99999343	1.00000196	1034.872
	FIM	1.00000003	1.00000004	2060.86
5 days	Iterative	1.00000301	1.00000817	1729.41
	FIM	1.00000003	1.00000003	3121.56

Table 2.3: Timing comparison for example 2

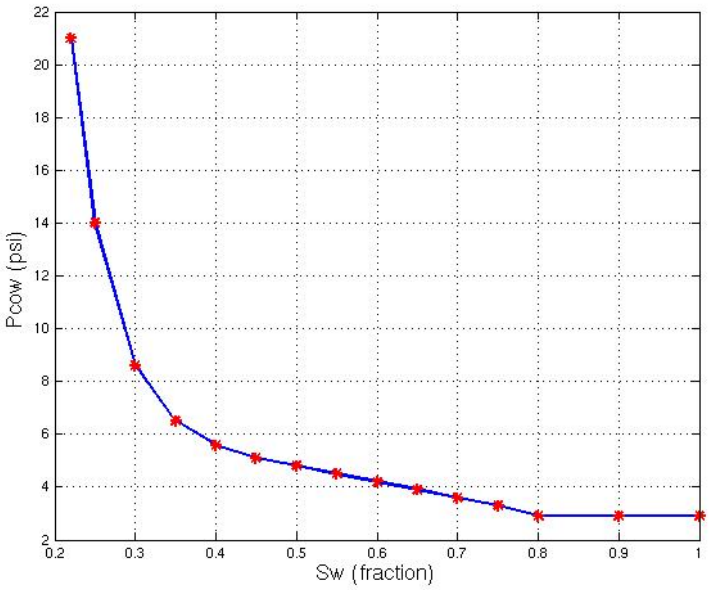

Figure 2.7: Capillary pressure curve for SPE 10

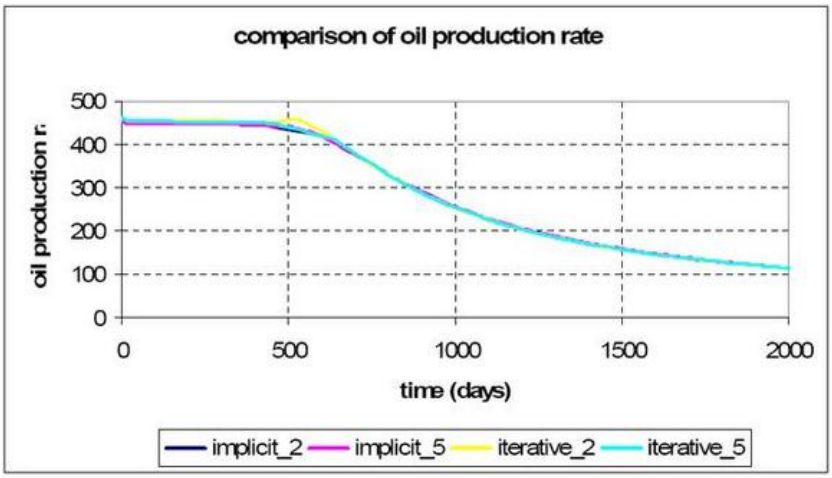

Figure 2.8: Oil production rate in producer 3 of example 2

clear as FIM in Figure 2.9 and 2.10.

2.6.3 Example 3: Coning Flow Problem

Oil production from a well that partly penetrates an oil zone overlying water may cause the oil/water interface to deform into a bell shape. This deformation is usually called water coning and occurs when vertical component of the viscous force exceeds the net gravity force [84, 101, 114]. At a certain production rate, the water cone will change from the stable equilibrium below the bottom of the well to instability and water breakthrough to the well occurs. This limiting rate is called the critical rate of water coning. Several models have been developed to define this critical value [59].

It is known that the IMPES method has difficulty simulating the water coning problem due to convergence failure. Here the iterative method is used to solve the water coning problem. The following example is modified from the SPE 2

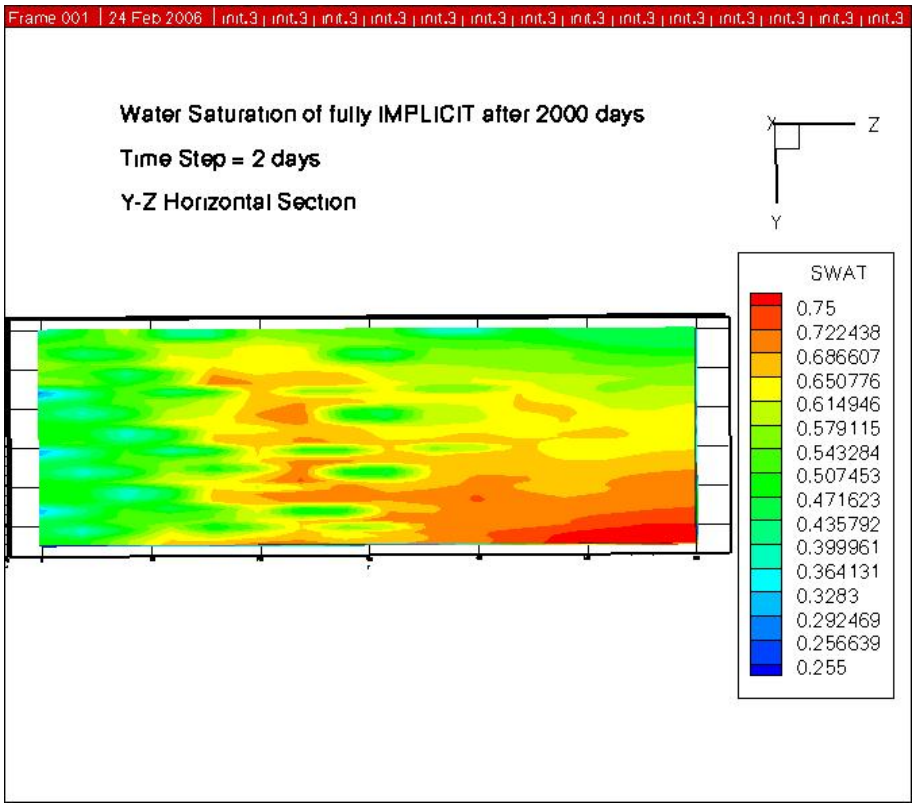

Figure 2.9: Saturation profile with the FIM after 2000 days

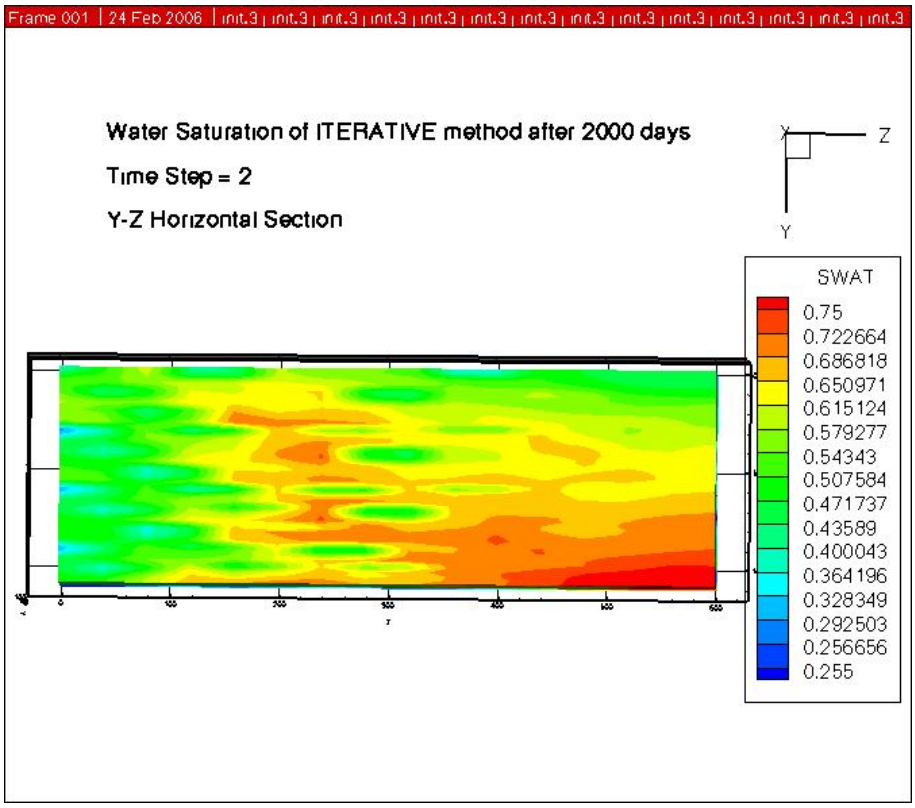

Figure 2.10: Saturation profile with the iterative model after 2000 days

Model	Oil balance	Water balance	CPU time (sec.)
FIM	1.0000009	0.9999917	88.45
Iterative	0.9999195	1.0000148	65.41

Table 2.4: Performance comparison for example 3

comparative project [121].

This model has 4335 grids, and the simulation was conducted for 500 days. For simplicity, the original radial grid has been changed to a rectangular grid. From the centre to outer boundary of the reservoir, the element dimension increases gradually. A production well is located at the center of the reservoir. The initial simulation time step size is 0.1 day, and the maximum time step size is 3 days.

As shown in Table 2.4, the iterative method requires $30 \sim 40\%$ less CPU time than the FIM for this coning problem. In addition, the iterative method achieves accuracy similar to the FIM, with material balance errors on the scale of 10^{-5}. We observed that the total number of time steps varies with different cases. Generally, for the same problem, the iterative method takes more time steps than the FIM. However, the simple structure of the iterative pressure equation decreases the CPU time per time step. More precisely, the decoupled system for the iterative model solves small subproblems and, as a result, it takes less CPU time than the larger FIM coupled system.

2.6.4 Example 4: Countercurrent Flow Problem

This example was chosen to show the ability of the iterative method to handle countercurrent flow problems. The reservoir is $600ft \times 600ft \times 400ft$, and the mesh grid is $30 \times 30 \times 20 = 18,000$. The upper subdomain of the reservoir with 10

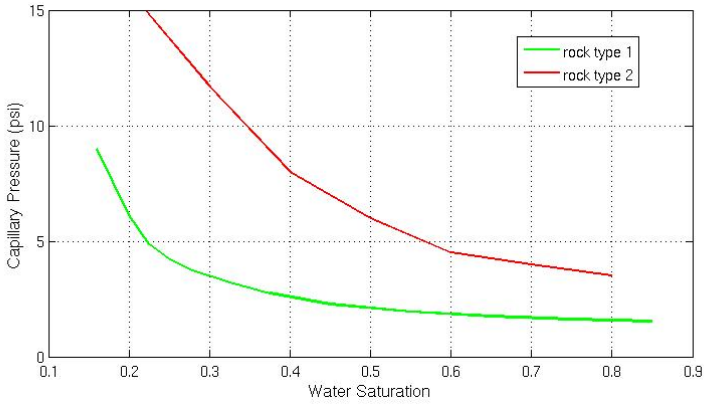

Figure 2.11: Capillary pressure curves for different rock types

layers has rock of type 1, which is more permeable than the rock of type 2 in the lower subdomain. Two different capillary pressure curves are applied on the two subdomains, as in Figure 2.11. An injection well and a production well are located at the two diagonal corners of the reservoir.

As shown in Figures 2.12 and 2.13, the water front in the upper subdomain flows faster than that in the lower subdomain. There is a high water content zone at the interface. During the simulation, water starts to flow downward, and oil starts to flow upward close to the interface. The water saturation profile reflects the water content in the reservoir after 400 days of simulation for both the FIM and the iterative method. The results of these two approaches are quite similar.

The total CPU time used in the iterative method case is 1733.23 seconds compared to 2547.28 seconds for the FIM. Thus, a reduction of more than 40% simulation time has been achieved by using the iterative method instead of the FIM. In addition, the iterative method reflects the countercurrent flow more accurately near the interface. This is another kind of problem that can not be simulated with IMPES.

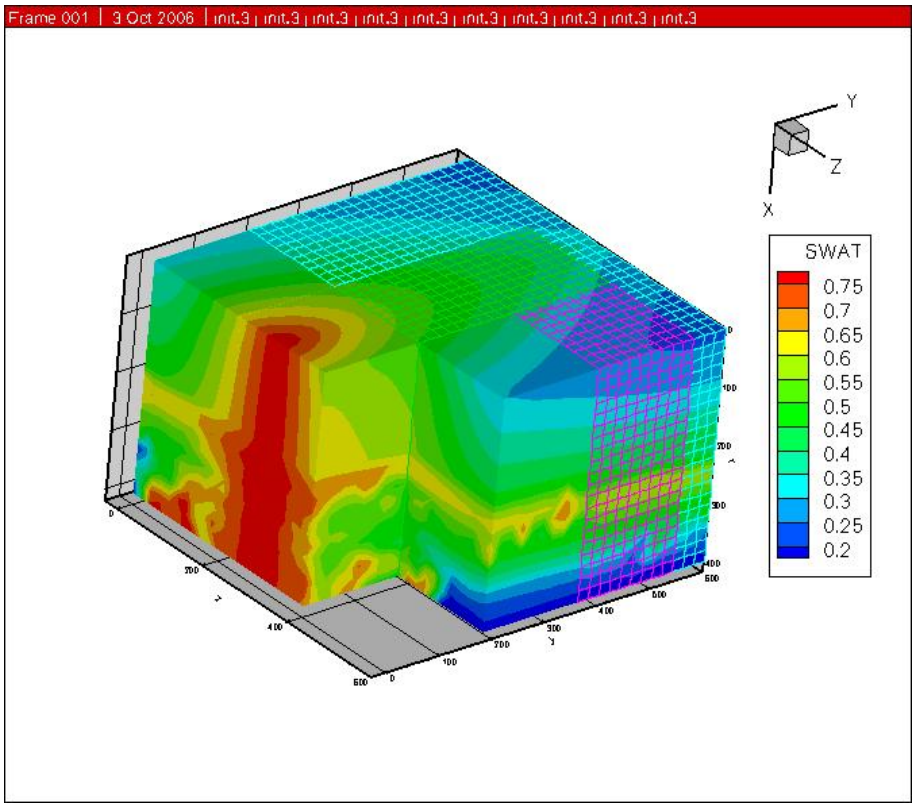

Figure 2.12: S_w after 400 days with the FIM

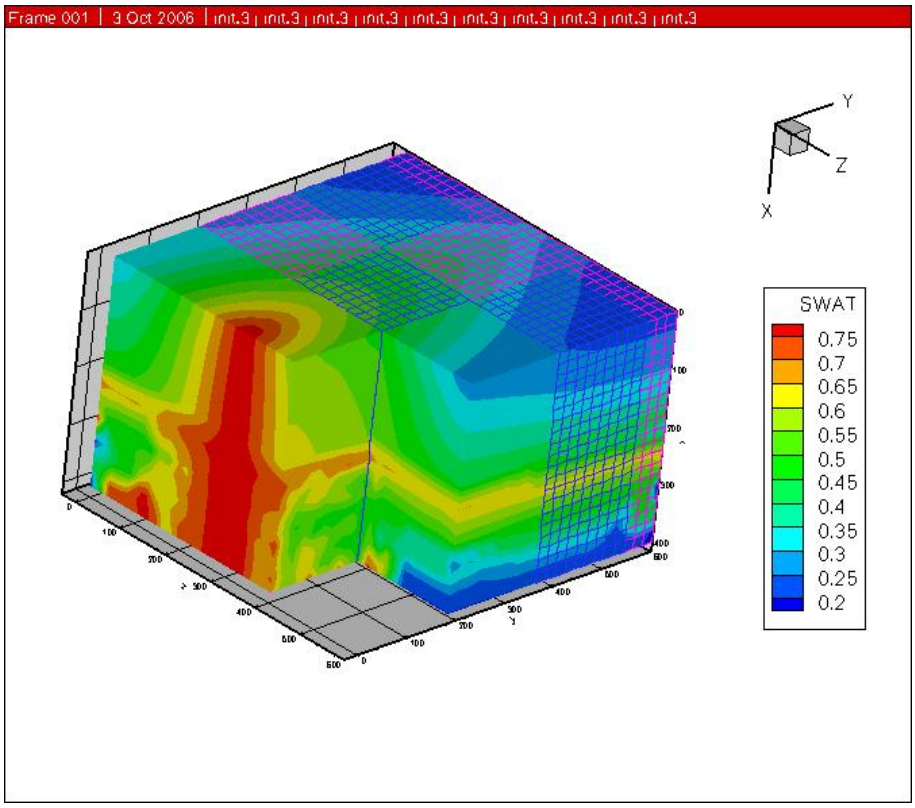

Figure 2.13: S_w after 400 days with the iterative method

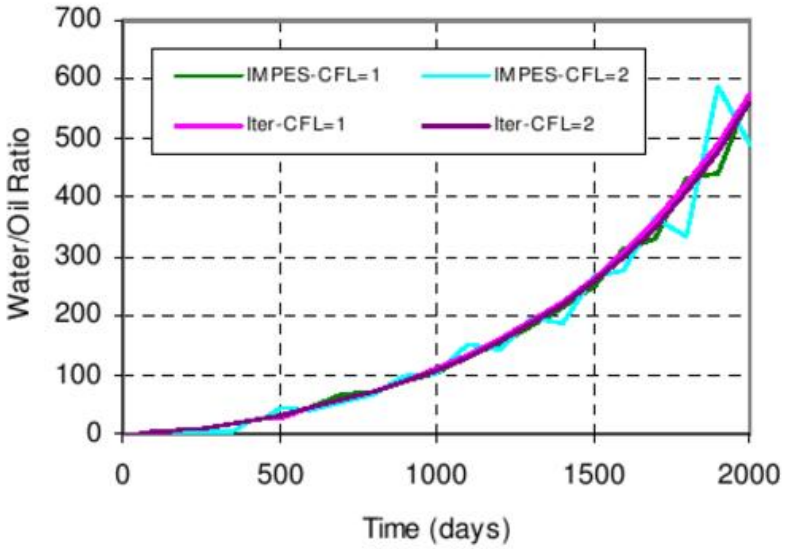

Figure 2.14: Water/oil ratio in the producer of example 5

2.6.5 Example 5: Stability Study

This example shows the stability of the iterative model and compares its performance with the IMPES method. The problem considered is a $3D$ heterogeneous reservoir with mesh $20 \times 20 \times 10 = 4000$; the permeability changes in vertical layers. Figure 2.14 shows the water/oil ratio in the producer for a 2000 day simulation. A CFL time step control has been applied with values 1 and 2 for both the IMPES and the iterative methods. The iterative method is stable for the entire simulation, but the IMPES method produces large oscillations at $CFL = 2$ after 1400 days. Even with small constraint, like $CFL = 1$, IMPES shows small oscillations after only 1700 days. In addition, we observe that the performance of IMPES degrades with increasing heterogeneity.

Figure 2.15 shows the results when the time step is set small enough to guar-

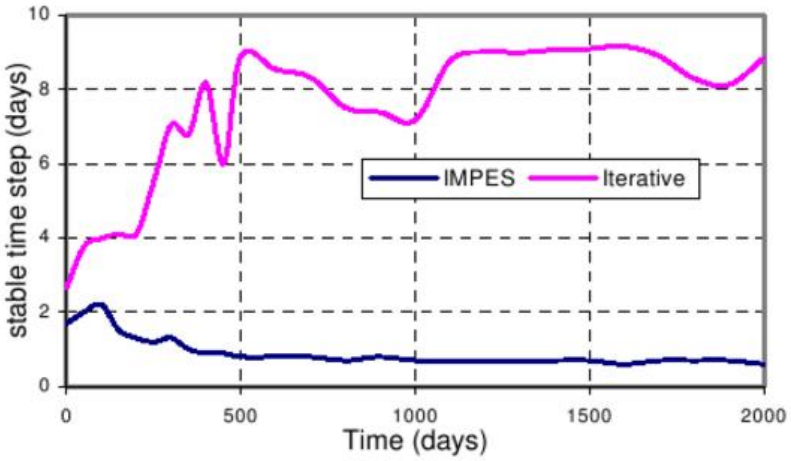

Figure 2.15: Maximum time step size with stable performance

antee the stability of both the IMPES and the iterative methods. In this specific case, the maximum time step for IMPES is less than 1 day, which is much smaller than the 9 day time step required by the iterative method to achieve stable performance. The high efficiency of these two methods comes with the decoupling of the whole equation system; however, the extremely small time step size is an obstacle to the effective applications of the IMPES method. The advantage in stability of the iterative method over the IMPES method has been shown clearly in this example.

2.6.6 Example 6: Air/Water Problem

In this section, an air/water reservoir problem is shown to validate the performance of the iterative air/water model. The reservoir has a mesh grid of $8 \times 32 \times 16 = 4096$. The reservoir has a low permeability profile, as shown in Figures 2.16 and 2.17. Two wells are located at opposite ends of the reservoir. The simulation results with time steps of 1 day and 2 days are listed in Table 2.5. Results show that, with the

Max time step	Model	Linear itns per step	Water balance	Air balance	CPU time (sec.)
1 days	FIM	294	0.9998769	0.9999451	267.5
	Iterative	597	1.000151	0.999217	165.7
2 days	FIM	402	0.9998124	0.9997831	317.5
	Iterative	708	1.000574	0.999004	195.4

Table 2.5: Timing comparison for the air/water model

iterative method, more than 30% of the CPU time has been saved, even though more iterations occur on one time step base. The material balance errors are on the scale of 10^{-4}. This error is quite acceptable for the air/water flow problem as it is very sensitive to fluid flow.

The contours in Figures 2.18 and 2.19 show air pressure and its gradient on the reservoir after 500 days of simulation. The iterative method produces solutions that are very close to those of the FIM.

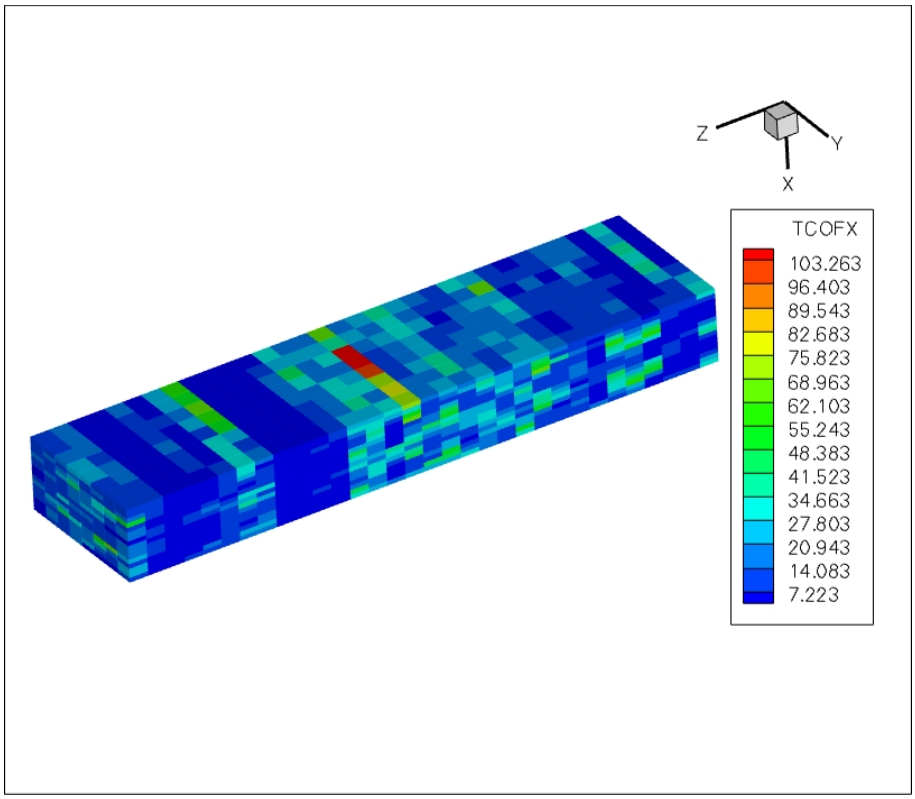

Figure 2.16: Reservoir vertical permeability for the air/water problem

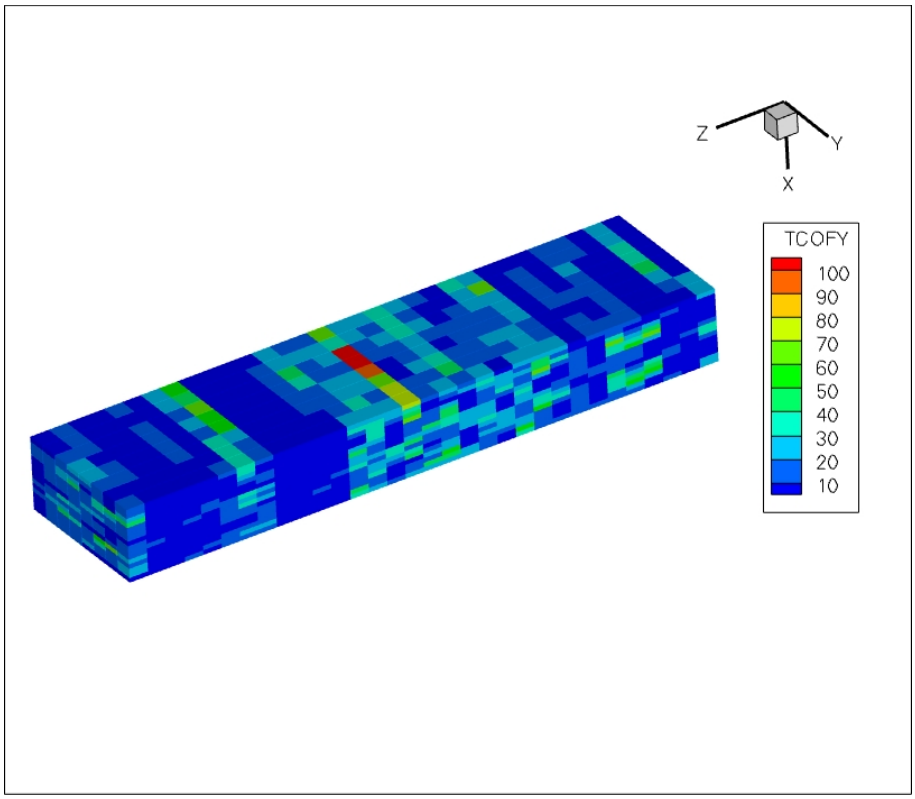

Figure 2.17: Reservoir horizontal permeability for the air/water problem

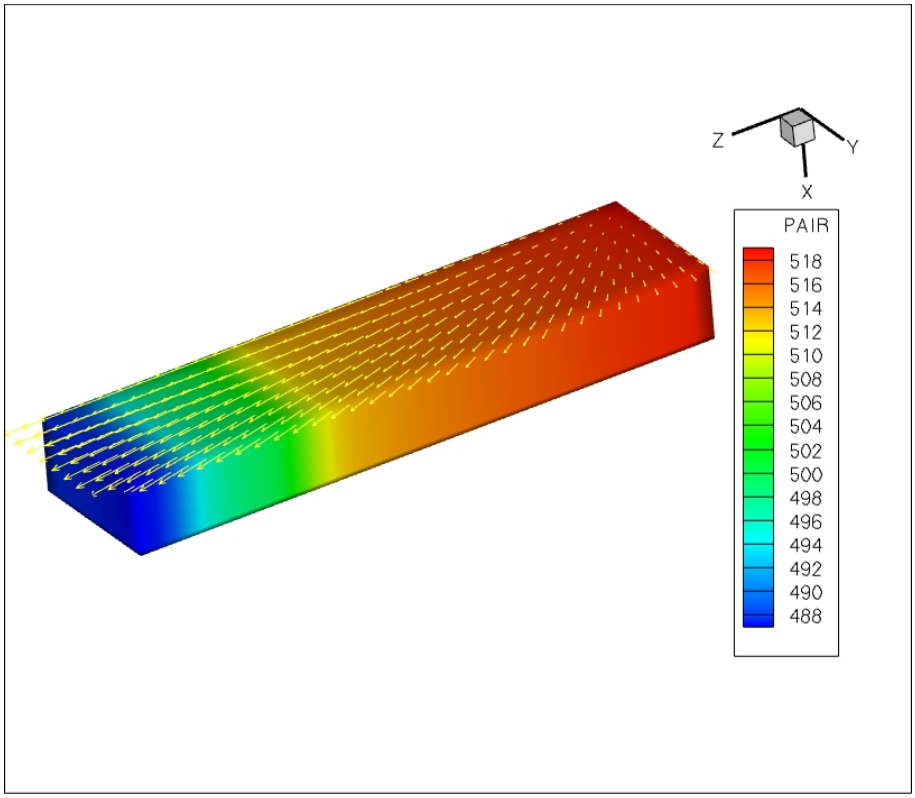

Figure 2.18: Air pressure after 500 days with the FIM

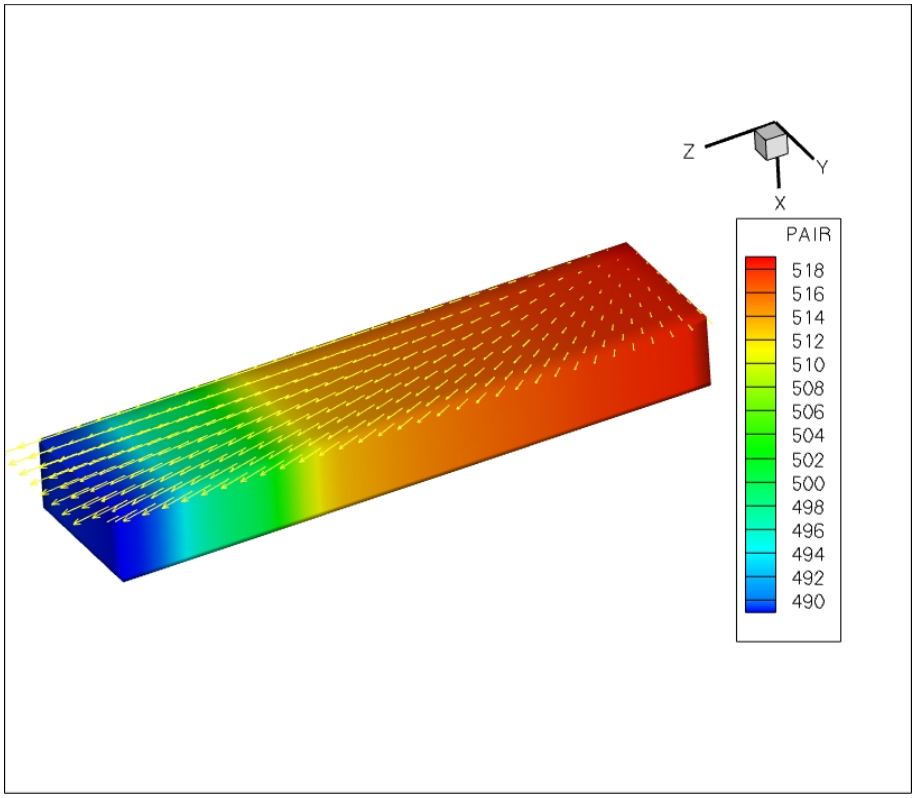

Figure 2.19: Air pressure after 500 days with the iterative method

Chapter 3

Iterative Black Oil Model

This chapter presents a development of the iterative black oil model and a study of its performance on various reservoir problems, including a comparison with the FIM. Again, this model is developed under the IPARS framework [77, 123].

3.1 Introduction

The iterative black oil model is developed like the iterative two-phase model. The following assumptions are made for the black oil model:

3.1.1 Assumption

- A three component system (W, O, G) with three phase (w, o, g).
- Oil and water are immiscible and no mass exchange occurs between these two phases.
- The gas component can only dissolve in the oil phase.

- Oil and water can not evaporate into the gas phase.
- Isothermal flow.
- Instantaneous thermodynamic equilibrium throughout the reservoir.
- The permeability tensor is diagonal and aligned with the coordinate system.
- Darcy's law for multiphase flow.
- No chemical reactions, precipitation or adsorption occur.
- The well is treated as either a source or a sink point.
- The formation is slightly compressible.

The differences between a *phase* and a *component* should be noted. In a hydrocarbon system, the *oil* component is the residual liquid at atmospheric pressure that is left after a differential vaporisation; the *gas* component can be mixed in liquid and gaseous states. The concept of *phase* is only meaningful for certain specific pressures and temperatures. For water, the *phase* and *component* concepts are the same. In the standard black oil model, the gas component exists in both the oil and the gas phases, but not in the water phase. No mass transfer occurs between the water phase and the other two phases.

3.1.2 Parameters

Most of the parameters for the iterative black oil model are defined as follows:

- Subscripts $w, o, and\, g$ refer to the water, oil, and gas phases, respectively.
- Subscripts $W, O, and\, G$ refer to water, oil, and gas components, respectively.

- $B_\alpha(P_o)$ is the formation volume factor as a function of the oil phase pressure P_o, which is equal to the ratio of the volume of phase α (including any dissolved components) under reservoir condition to the stock tank volume of phase α $[bbl/stb]$.
- $b_\alpha = 1/B_\alpha$.
- c_α = phase compressibility $[psi^{-1}]$.
- K = diagonal permeability tensor $[md]$.
- $k_{r\alpha}$ = relative permeability for phase α.
- q_M = mass rate of component M, $M = W, O, G$ [scf/d for water and oil, $mscf/d$ for gas].
- q_{MS} = mass rate of component M at standard condition.
- $R_{so}(P_O)$ = the solution gas-oil ratio as a function of P_o, equal to the stock tank volume of gas component dissolved in a stock tank volume of oil component when the oil is saturated with gas $[mscf/scf]$.
- U_M = mass flow rate of component M through a unit area $scf/ft^2 - day$ for water and oil, $mscf/ft^2 - day$ for gas components.
- $\phi(P_o)$ = porosity as a function of P_o.
- μ_α = viscosity of phase $\alpha[cp]$.
- λ_α = mobility of phase α, equal to $k_{r\alpha}/\mu_\alpha$.
- ρ_α = density of phase α $[lb/cu - ft]$.

- ρ_{MS} = stock tank density of component M [lb/scf for water and oil components and $lb/mscf$ for gas component].
- S_α = saturation for phase α.
- N_M = the concentration of component M; equal to the product of density and saturation within all related phases.

3.1.3 Variable Relations

Several relations must be defined to relate the concepts of *phase* and *component* in the black oil system. They are listed below.

Density

$\rho_W, \rho_O, and\, \rho_G$ are component densities; $\rho_w, \rho_o, and\, \rho_g$ are phase densities; and the subscript S stands for standard conditions at the surface.

- For the water phase/component,

$$\rho_W = \rho_w = \frac{\rho_{WS}}{B_w}. \tag{3.1}$$

- For the free gas phase,

$$\rho_g = \frac{\rho_{GS}}{B_g}. \tag{3.2}$$

- Oil phase is a mixture of the oil and gas components,

$$\rho_o = \frac{\rho_{OS} + R_{so}\rho_{GS}}{B_o}. \tag{3.3}$$

Component Concentration

- For the water component,

$$N_W = \rho_W S_w = \rho_w S_w = \frac{\rho_{WS} S_w}{B_w}. \tag{3.4}$$

- For the oil component,

$$N_O = \rho_O S_o = \frac{\rho_{OS} S_o}{B_o}. \tag{3.5}$$

- The gas component includes two parts; one is free gas and the other dissolves in oil. So both gas and oil formation volume factors are involved here:

$$N_G = \rho_G S_g + \rho'_G R_{so} S_o = \frac{\rho_{GS} S_g}{B_g} + \frac{\rho_{GS} R_{so} S_o}{B_o}, \tag{3.6}$$

where $\rho'_G = \frac{\rho_{GS}}{B_{so}}$ is the density of dissolved gas in the oil phase.

3.2 Numerical Formulation

3.2.1 Pressure Equation

Mass conservation equations for each component (W, O, G)

:

$$\begin{aligned}
\frac{\partial(\phi N_W)}{\partial t} &= -\nabla \cdot (\rho_W u_w) + q_W && (3.7)\\
\frac{\partial(\phi N_O)}{\partial t} &= -\nabla \cdot (\rho_O u_o) + q_O && (3.8)\\
\frac{\partial(\phi N_G)}{\partial t} &= -\nabla \cdot (\rho_G u_g + R_{so} \rho'_G u_o) + q_G, && (3.9)
\end{aligned}$$

where

$$\begin{aligned} q_W &= \frac{\rho_{WS}}{B_w} q_{WS} \\ q_O &= \frac{\rho_{OS}}{B_o} q_{OS} \\ q_G &= \frac{\rho_{GS}}{B_g} q_{GS} + R_{so} \frac{\rho_{GS}}{B_o} q_{OS}. \end{aligned} \tag{3.10}$$

Darcy flow

Darcy's law for multiphase flow is used to calculate the volumetric velocity of each phase:

$$u_w = -K \frac{k_{rw}}{\mu_w} \left(\nabla P_w - \rho_w g \nabla D \right) \tag{3.11}$$

$$u_o = -K \frac{k_{ro}}{\mu_o} \left(\nabla P_o - \rho_o g \nabla D \right) \tag{3.12}$$

$$u_g = -K \frac{k_{rg}}{\mu_g} \left(\nabla P_g - \rho_g g \nabla D \right). \tag{3.13}$$

Here we define phase mobility as $\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}$. $P_o, N_w, and\, N_o$ are chosen as the primary variables. Again, the saturation constraint equation and capillary pressure equations are applied with the following formula:

$$\begin{aligned} S_w + S_o + S_g = 1, \\ P_{cow} = P_o - P_w, \\ P_{cgo} = P_g - P_o. \end{aligned} \tag{3.14}$$

In the following derivation of the iterative method, we first substitute Darcy

velocity equations into the mass conservation equations. In addition we recall the phase scaling that we applied in the iterative air/water model to achieve significant improvements. It is extended here as well, for the black oil model. However, following exactly what was done in the two-phase model will lead to difficulties due to the mass transfer between the gas and oil phases in the black oil system. For simplicity, all three equations are thus multiplied by the phase viscosities $\mu_w, \mu_o, and\, \mu_g$, and then divided by stock tank densities $q_{WS}, q_{OS}, and\, q_{GS}$, respectively, yielding

$$\frac{\partial}{\partial t}\left(\phi\mu_w\frac{S_w}{B_w}\right) = \nabla\cdot\left[K\frac{k_{rw}}{B_w}\left(\nabla P_w - \rho_w g\nabla D\right)\right] + \mu_w\frac{q_{WS}}{B_w} \tag{3.15}$$

for water,

$$\frac{\partial}{\partial t}\left(\phi\mu_o\frac{S_o}{B_{so}}\right) = \nabla\cdot\left[K\frac{k_{ro}}{B_{so}}\left(\nabla P_o - \rho_o g\nabla D\right)\right] + \mu_o\frac{q_{OS}}{B_{so}} \tag{3.16}$$

for oil, and

$$\begin{aligned}\frac{\partial}{\partial t}\left[\phi\mu_g\left(\frac{S_g}{B_g} + R_{so}\frac{S_o}{B_{so}}\right)\right] &= \nabla\cdot\left[K\frac{k_{rg}}{B_g}\left(\nabla P_g - \rho_g g\nabla D\right)\right] \\ +\nabla\cdot\left[KR_{so}\mu_g\frac{\lambda_o}{B_{so}}\left(\nabla P_o - \rho_o g\nabla D\right)\right] &+ \mu_g\frac{q_{GS}}{B_g} + R_{so}\mu_g\frac{q_{OS}}{B_{so}}\end{aligned} \tag{3.17}$$

for gas.

Substituting Eq.(3.14) into Eq.(3.15), we obtain the following water phase equation

$$\begin{aligned}&\frac{\partial}{\partial t}\left(\phi\mu_w\frac{S_w}{B_w}\right) \\ =\;& \nabla\cdot\left(K\frac{k_{rw}}{B_w}\nabla P_o\right) - \nabla\cdot\left[K\frac{k_{rw}}{B_w}\left(\nabla P_{cow} + \rho_w g\nabla D\right)\right] + \mu_w\frac{q_{WS}}{B_w},\end{aligned} \tag{3.18}$$

and similarly substituting Eq.(3.14) into Eq.(3.17) we have

$$\begin{aligned}
&\frac{\partial}{\partial t}\left[\phi\mu_g\left(\frac{S_g}{B_g}+R_{so}\frac{S_o}{B_{so}}\right)\right]=\nabla\cdot\left[K\left(\frac{k_{rg}}{B_g}+R_{so}\mu_g\frac{\lambda_o}{B_{so}}\right)\nabla P_o\right]\\
+\quad&\nabla\cdot\left[K\frac{k_{rg}}{B_g}\left(\nabla P_{cgo}-\rho_g g\nabla D\right)-KR_{so}\mu_g\frac{\lambda_o}{B_{so}}\rho_o g\nabla D\right]\\
+\quad&\mu_g\frac{q_{GS}}{B_g}+R_{so}\mu_g\frac{q_{OS}}{B_{so}}.
\end{aligned}\tag{3.19}$$

The oil equation remains as in Eq.(3.16). We continue with Eq.(3.16), Eq.(3.18), and Eq.(3.19) to derive a pressure equation for the iterative black oil model. Considering that $B_w, B_o, B_g, R_{so}, and\, \phi$ are functions of oil pressure P_o, all of them contribute to the construction of the coefficient matrix for the pressure equation. All other variables that are functions of concentration/saturation are upwinded from the values in the last iteration.

In the following derivations, k is the iteration number of the current time step $n+1$; n represents the number of the last time step.

For storage terms in Eq.(3.18), we use the following expression:

$$\begin{aligned}
\frac{\partial}{\partial t}\left(\phi\mu_w\frac{S_w}{B_w}\right)&=\mu_w\frac{\left(\phi\mu_g\frac{S_w}{B_w}\right)^{k+1}-\left(\phi\frac{S_w}{B_w}\right)^{n}}{\triangle t^{n+1}}\\
&=\mu_w\frac{\left(\phi\frac{S_w}{B_w}\right)^{k}-\left(\phi\frac{S_w}{B_w}\right)^{n}+\delta\left(\phi\frac{S_w}{B_w}\right)}{\triangle t^{n+1}}.
\end{aligned}\tag{3.20}$$

Similarly, for the oil and gas phases:

$$\frac{\partial}{\partial t}\left(\phi\mu_o\frac{S_o}{B_{so}}\right)=\mu_o\frac{\left(\phi\frac{S_o}{B_{so}}\right)^{k}-\left(\phi\frac{S_o}{B_{so}}\right)^{n}+\delta\left(\phi\frac{S_o}{B_{so}}\right)}{\triangle t^{n+1}}\tag{3.21}$$

$$\begin{aligned}
&\frac{\partial}{\partial t}\left[\phi\mu_g\left(\frac{S_g}{B_g}+R_{so}\frac{S_o}{B_{so}}\right)\right] \\
= \quad & \mu_g\frac{\left[\phi\left(\frac{S_g}{B_g}+R_{so}\frac{S_o}{B_{so}}\right)\right]^k-\left[\phi\left(\frac{S_g}{B_g}+R_{so}\frac{S_o}{B_{so}}\right)\right]^n+\delta\left[\phi\left(\frac{S_g}{B_g}+R_{so}\frac{S_o}{B_{so}}\right)\right]}{\triangle t^{n+1}}
\end{aligned} \tag{3.22}$$

In Eq.(3.20), for water,

$$\delta\left(\phi\mu_w\frac{S_w}{B_w}\right)=\mu_w\left[\left(\frac{S_w}{B_w}\right)\left(\frac{d\phi}{dP_o}\right)+(\phi S_w)\left(\frac{dB_w^{-1}}{dP_o}\right)\right]^k\delta P_o^{k+1}. \tag{3.23}$$

In Eq.(3.21), for oil,

$$\delta\left(\phi\mu_o\frac{S_o}{B_{so}}\right)=\mu_o\left[\left(\frac{S_o}{B_{so}}\right)\left(\frac{d\phi}{dP_o}\right)+(\phi S_o)\left(\frac{dB_{so}^{-1}}{dP_o}\right)\right]^k\delta P_o^{k+1}. \tag{3.24}$$

In Eq.(3.22), for gas

$$\begin{aligned}
&\delta\left[\phi\mu_g\left(\frac{S_g}{B_g}+R_{so}\frac{S_o}{B_o}\right)\right] \\
= \quad & \mu_g\left[\left(\frac{S_g}{B_g}+R_{so}\frac{S_o}{B_o}\right)\left(\frac{d\phi}{dP_o}\right)+(\phi S_g)\left(\frac{dB_g^{-1}}{dP_o}\right)\right]^k\delta P_o^{k+1} \\
+ \quad & \mu_g\left[(\phi R_{so}S_o)\left(\frac{dB_o^{-1}}{dP_o}\right)+\left(\phi\frac{S_o}{B_{so}}\right)\left(\frac{dR_{so}}{dP_o}\right)\right]^k\delta P_o^{k+1}.
\end{aligned} \tag{3.25}$$

For simplicity, $C_{1w}, C_{1o}, and\, C_{1g}$ are used to represent the coefficients in Eq.(3.23)-Eq.(3.25) respectively. Eq.(3.16), (3.18) and Eq.(3.19) are summed and coefficients are substituted. Oil pressure terms are combined on the left-hand-side of the equation and other terms remain to the right-hand-side. The pressure equation of the

iterative black oil model can be written as

$$\begin{aligned}
& \nabla \cdot \left[K \left(\frac{k_{rw}}{B_w} + \frac{k_{ro}}{B_{so}} + \frac{k_{rg}}{B_g} + R_{so}\mu_g \frac{\lambda_o}{B_{so}} \right)^k \nabla P_o^{k+1} \right] \\
- & \frac{(C_{1w} + C_{1o} + C_{1g})^k}{\triangle t^{n+1}} \delta P_o^{k+1} + \mu_w \frac{q_{WS}}{B_w} + \mu_g \frac{q_{GS}}{B_g} + (\mu_o + \mu_g R_{so}) \frac{q_{OS}}{B_{so}} \\
= & (D_{1w} + D_{1o} + D_{1g})^k + D_{pcw}^k - D_{pcg}^k + (D_{2w} + D_{2o} + D_{2g})^k , \qquad (3.26)
\end{aligned}$$

where

$$\begin{aligned}
D_{1w}^k &= \mu_w \frac{\left(\phi \frac{S_w}{B_w}\right)^k - \left(\phi \frac{S_w}{B_w}\right)^n}{\triangle t^{n+1}}, \\
D_{1o}^k &= \mu_o \frac{\left(\phi \frac{S_o}{B_{so}}\right)^k - \left(\phi \frac{S_o}{B_{so}}\right)^n}{\triangle t^{n+1}}, \\
D_{1g}^k &= \mu_g \frac{\left[\phi \left(\frac{S_g}{B_g} + R_{so}\frac{S_o}{B_{so}}\right)\right]^k - \left[\phi \left(\frac{S_g}{B_g} + R_{so}\frac{S_o}{B_{so}}\right)\right]^n}{\triangle t^{n+1}}, \\
D_{pcw}^k &= \nabla \cdot \left(K \frac{k_{rw}}{B_w} \nabla P_{cow} \right)^k , \\
D_{pcg}^k &= \nabla \cdot \left(K \frac{k_{rg}}{B_g} \nabla P_{cgo} \right)^k , \\
D_{2w}^k &= \nabla \cdot \left(K \frac{k_{rw}}{B_w} \rho_w g \nabla D \right)^k , \\
D_{2o}^k &= \nabla \cdot \left(K \frac{k_{ro}}{B_{so}} \rho_o g \nabla D \right)^k , \\
D_{2g}^k &= \nabla \cdot \left[K \left(\frac{k_{rg}}{B_g} \rho_g + R_{so}\mu_g \frac{\lambda_o}{B_{so}} \right) g \nabla D \right]^k . \qquad (3.27)
\end{aligned}$$

For the two-phase situation with $N_G < R_{so}N_O$ and $S_g = 0$, certain simplifications are made to the three-phase formula, such that all the terms with gas phase

disappear:

$$\begin{aligned} R_o &= \frac{N_G}{N_O}, \\ B_o &= B_{do} + (B^*_{so} - B^*_{do})\,\frac{R_o}{R_{so}}, \\ \rho_o &= \frac{\rho_{OS} + R_o\rho_{GS}}{B_o}, \\ N_O &= \frac{\rho_{OS} S_o}{B_o}, \\ S_w &= 1 - S_o, \\ N_W &= (1 - B_o N_O)\, b_w. \end{aligned}$$

The following is the resulting pressure equation:

$$\begin{aligned} & \nabla \cdot \left[K \left(\frac{k_{rw}}{B_w} + \frac{k_{ro}}{B_o} + R_o \mu_g \frac{\lambda_o}{B_o} \right)^k \nabla P_o^{k+1} \right] \\ - & \frac{(C_{1w} + C_{1o} + C_{1g})^k}{\triangle t^{n+1}} \delta P_o^{k+1} + \mu_w \frac{q_{WS}}{B_w} + (\mu_o + \mu_g R_o)\,\frac{q_{OS}}{B_o} \\ = & (D_{1w} + D_{1o} + D_{1g})^k + D^k_{pcw} + (D_{2w} + D_{2o} + D_{2g})^k\,. \end{aligned} \tag{3.28}$$

Some of the initial parameters are changed accordingly,

$$\begin{aligned} D^k_{1w} &= \mu_w \frac{\left(\phi \frac{S_w}{B_w}\right)^k - \left(\phi \frac{S_w}{B_w}\right)^n}{\triangle t^{n+1}}, \\ D^k_{1o} &= \mu_o \frac{\left(\phi \frac{S_o}{B_o}\right)^k - \left(\phi \frac{S_o}{B_o}\right)^n}{\triangle t^{n+1}}, \\ D^k_{1g} &= \mu_g \frac{\left[\phi R_o \frac{S_o}{B_o}\right]^k - \left[\phi R_o \frac{S_o}{B_o}\right]^n}{\triangle t^{n+1}}, \\ D^k_{pcw} &= \nabla \cdot \left(K \frac{k_{rw}}{B_w} \nabla P_{cow} \right)^k, \\ D^k_{2w} &= \nabla \cdot \left(K \frac{k_{rw}}{B_w} \rho_w g \nabla D \right)^k, \end{aligned}$$

$$
\begin{aligned}
D_{2o}^{k} &= \nabla \cdot \left(K \frac{k_{ro}}{B_o} \rho_o g \nabla D \right)^{k}, \\
D_{2g}^{k} &= \nabla \cdot \left[K R_o \mu_g \frac{\lambda_o}{B_o} g \nabla D \right]^{k}.
\end{aligned} \tag{3.29}
$$

3.2.2 Concentration Equation

Once oil pressure has been obtained, mass conservation equations are used to calculate oil and water concentrations. Before solving for concentration, all parameters related to pressure are updated, and all other parameters are upwinded from the last iteration:

$$
N_O^{k+1} = \frac{\triangle t^{n+1} \left(-\nabla \cdot \left(\rho_O^{k+1} u_o^{k+1} \right) + q_O^{k+1} \right) + \phi^n N_O^n}{\phi^{k+1}}, \tag{3.30}
$$

$$
N_W^{k+1} = \frac{\triangle t^{n+1} \left(-\nabla \cdot \left(\rho_W^{k+1} u_w^{k+1} \right) + q_W^{k+1} \right) + \phi^n N_W^n}{\phi^{k+1}}. \tag{3.31}
$$

3.2.3 Well Term

For the well term, the same formula used in two-phase model is implemented as follows:

$$
q_\alpha = \frac{GLKk_{r\alpha}}{\mu_\alpha} \left(P_{wb} - P_\alpha \right), \tag{3.32}
$$

where P_{wb} is the wellbore pressure; P_α is the pressure of phase α in the grid block; K is the absolute permeability; and L is the length of the open wellbore penetrating on the grid cell. G is the dimensionless geometric factor given by

$$
G = \frac{2\pi}{ln\left(\frac{r_{eq}}{r_w}\right) + s}, \tag{3.33}
$$

where s is the skin factor of the wellbore; r_w is the wellbore radius (default is $0.5inch$); and r_{eq} is the equivalent radius of the grid cell. According to Peaceman's model[86], r_{eq} is given by

$$r_{eq} = 0.28 \frac{\left[(\triangle x)^2 \left(\frac{k_y}{k_x}\right)^{1/2} + (\triangle y)^2 \left(\frac{k_x}{k_y}\right)^{1/2} \right]^{1/2}}{(\ frack_y k_x)^{1/4} + \left(\frac{k_x}{k_y}\right)^{1/4}}, \tag{3.34}$$

where $\triangle x$ and $\triangle y$ are in the x and y direction of the reservoir, respectively; k_x and k_y are reservoir directional permeabilities. The well term is treated implicitly in the pressure equations.

3.2.4 Relative Permeability

The relative permeability of two-phase flow can be obtained through lab experiments. The water and gas relative permeabilities in three-phase flow are the same as in two-phase flow. However, the oil relative permeability in three-phase flow must be calculated based on two-phase permeability values. There are several methods available to calculate this three-phase oil relative permeability and, three of them are used for the iterative model in IPARS [64, 77, 123], namely Stone's model I, Stone's model II, and the Eclipse-100 model [38].

For most of the following numerical experiments with the iterative black oil model, Stone's Model II is the default option. It is given by

$$k_{ro} = k^0_{row} \left[\left(\frac{k_{row}}{k^0_{row}} + k_{rw} \right) \left(\frac{k_{rog}}{k^0_{row}} + k_{rg} \right) - (k_{rw} + k_{rg}) \right], \tag{3.35}$$

where k^0_{row} is the endpoint relative permeability of the oil phase at residual water saturation. All negative values of k_{ro} are set to 0 to make the simulation reasonable.

3.3 Convergence Control

After solving for the concentration, saturation and other fluid properties are updated and mass balance errors are calculated both globally and locally. the current time step finishes once the convergence criteria are satisfied. Otherwise, an additional iteration is conducted. Changes in parameters such as saturation can also be used as convergence control criteria.

The extrapolation and forcing functions applied in the two-phase model are also adopted here. The simulation flow of the iterative black oil model is illustrated in Figure 3.1.

3.4 Numerical Examples

3.4.1 Oxbow Problem

This numerical example presents a large oxbow reservoir of dimension $37ft \times 2050ft \times 1106ft$ with 125,385 total grid elements. 2 injectors are located at the head of the oxbow and 7 producers are located in other areas of the reservoir. All of the wells are bottom hole pressure specified.

Both the FIM and the iterative method are used in this example, and the results are listed in Table 3.1. If a time step of 2 days is used, the iterative method reduces the simulation time by around 30%; if a time step of 5 days is used, the iterative method reduces the simulation time by around 20%. In addition, the mass balance errors are on the scale of 10^{-4}. This case shows that the iterative black oil model is much faster than the FIM with reasonably accurate results.

Figures $3.2 - 3.5$ illustrate water saturation and oil pressure after 200 days

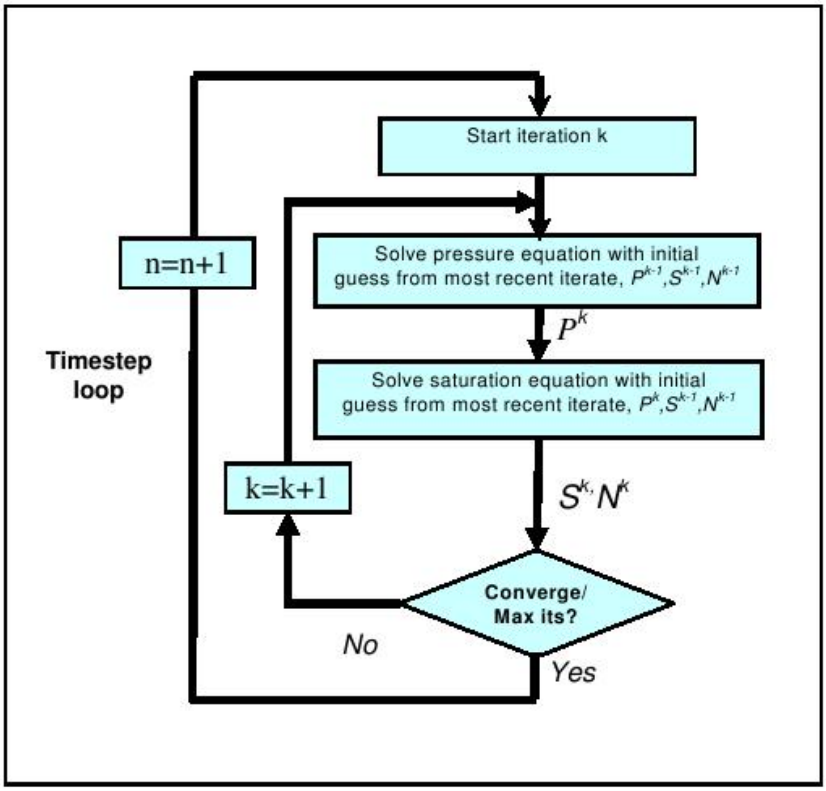

Figure 3.1: Flow chart for the iterative black oil model

Max time step	Model	Oil balance	Water balance	Gas balance	CPU time (sec.)
2 days	Iterative	1.0001653	0.9998192	1.0003765	11061
	FIM	1.0000042	0.9999994	1.0000117	15630
5 days	Iterative	0.9996115	0.9990412	0.9992377	10057
	FIM	0.9999966	0.9999982	0.9999978	12376

Table 3.1: Timing comparison for the large oxbow problem

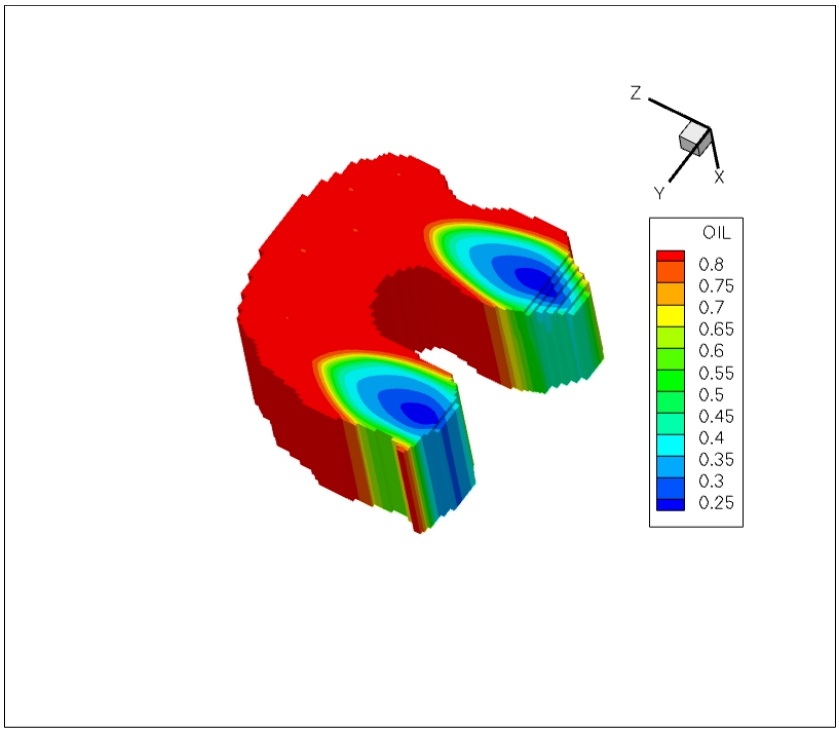

Figure 3.2: S_w after 200 days with the FIM for example 1

simulation for both the iterative method and the FIM. The results of these two methods match very well.

3.4.2 Upscaled SPE 10

This example is an upscaled SPE 10 reservoir that is the same as in Section 2.6.2. The vertical and horizontal permeabilities, in log scale, are shown in Figure 3.6 and Figure 3.7, respectively. Since SPE 10 is originally designed for two-phase simulation, the PVT data and additional physical properties for three-phase are modified from SPE 9. This case is run using the FIM and the iterative methods; Eclipse100 is also used for the purpose of comparing accuracy and efficiency. Figure

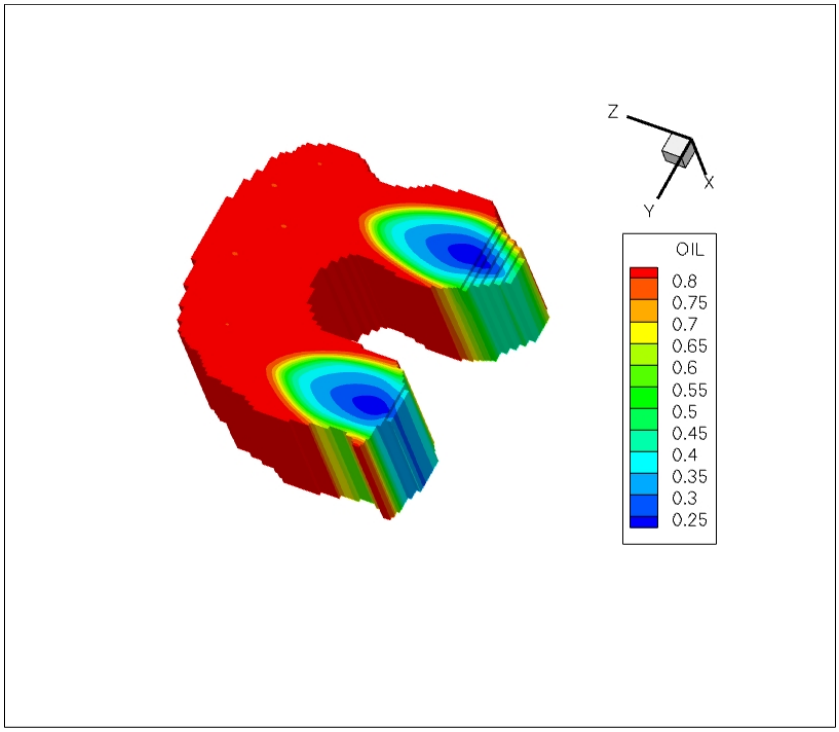

Figure 3.3: S_w after 200 days with the iterative method for example 1

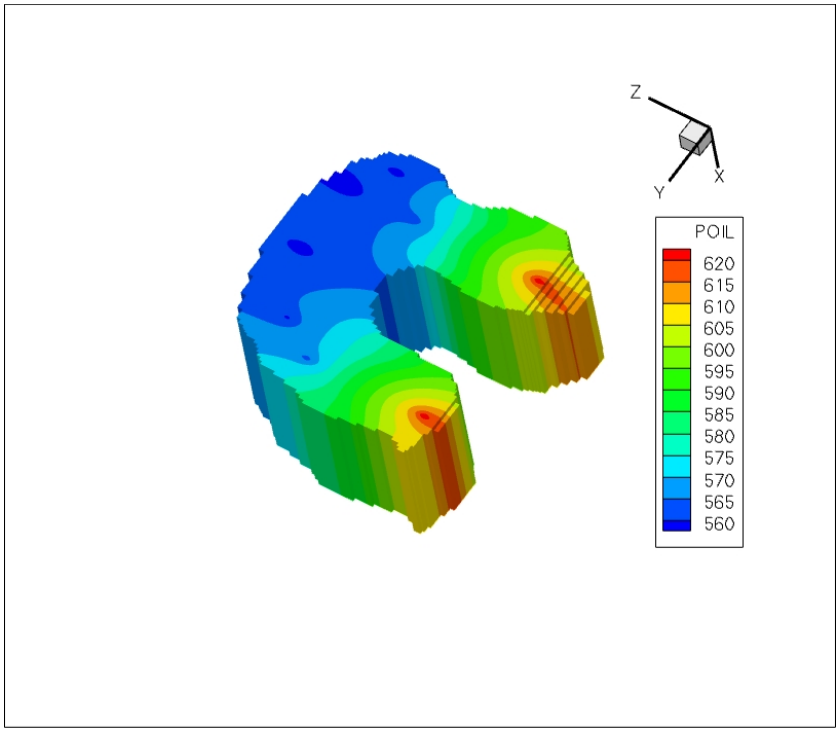

Figure 3.4: P_{oil} after 200 days with the FIM for example 1

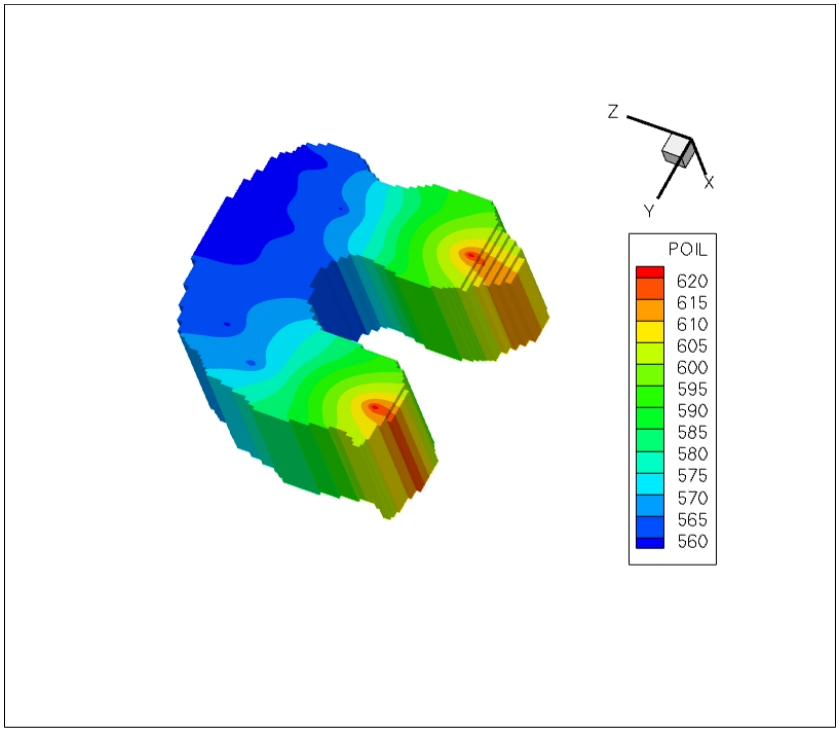

Figure 3.5: P_{oil} after 200 days with the iterative method for example 1

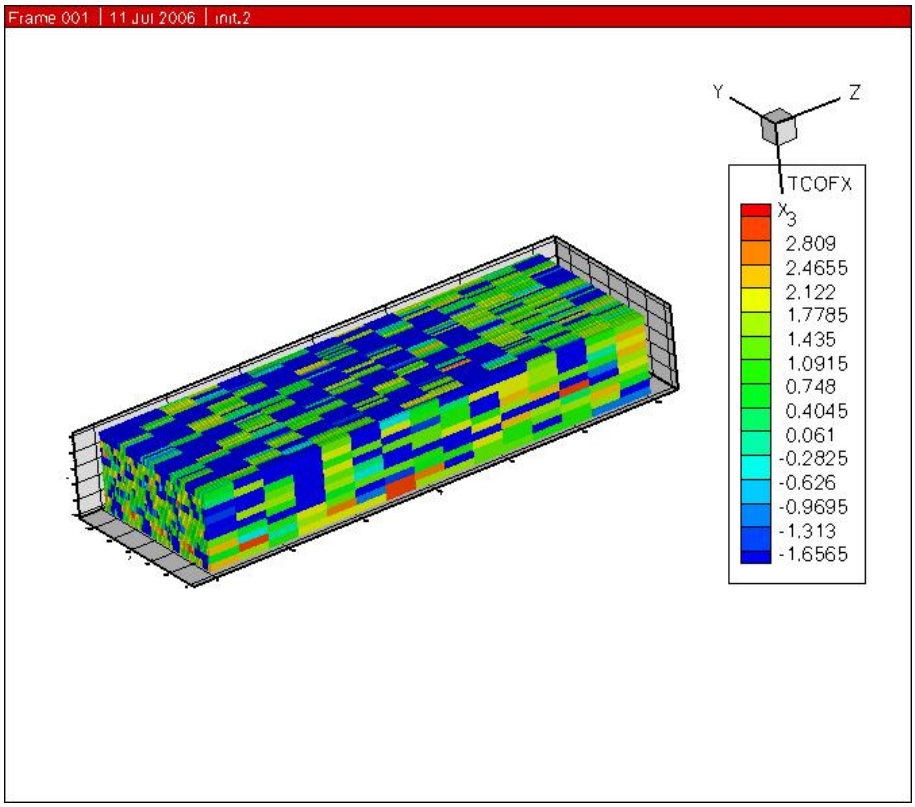

Figure 3.6: Reservoir vertical permeability for example 2

3.8 shows the oil production rate in one of the four producers. The iterative method produces a small difference from the other two methods at the beginning, but it approaches the solution of the FIM and Eclipse100 very well in later time.

The timing performance of the iterative method and the FIM are listed in Table 3.2. In this case, the iterative method produces better results than in example 1 in that total CPU time has been reduced by 35% when the time step was set to either 2 days or 5 days. The mass balance errors and the matching quality of the water/oil ratio in producer 3 in Figure 3.9 indicate accurate results with the iterative method, which is very close to the result of the FIM.

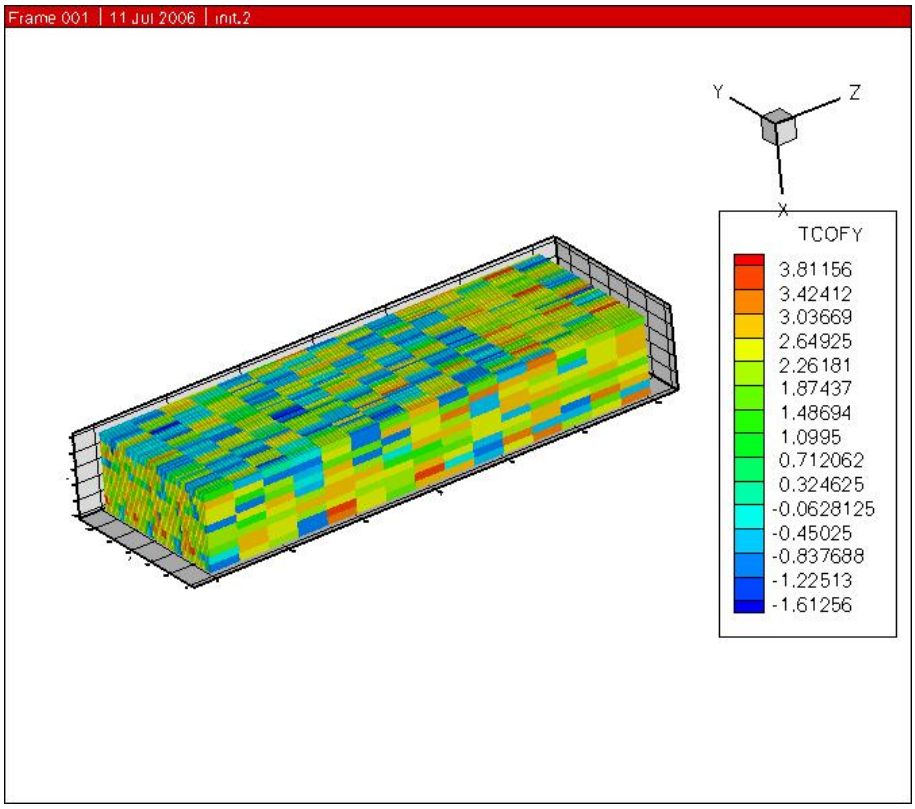

Figure 3.7: Reservoir horizontal permeability for example 2

Max time step	Model	Oil balance	Water balance	Gas balance	CPU time (sec.)
2 days	Iterative	1.0002085	0.9998015	1.0006430	1917
	FIM	1.0000006	0.9999994	1.0000012	3060
5 days	Iterative	0.9995802	0.9991362	0.9990452	1840
	FIM	0.9999989	0.9999997	0.9999986	2867

Table 3.2: Timing comparison for upscaled SPE 10

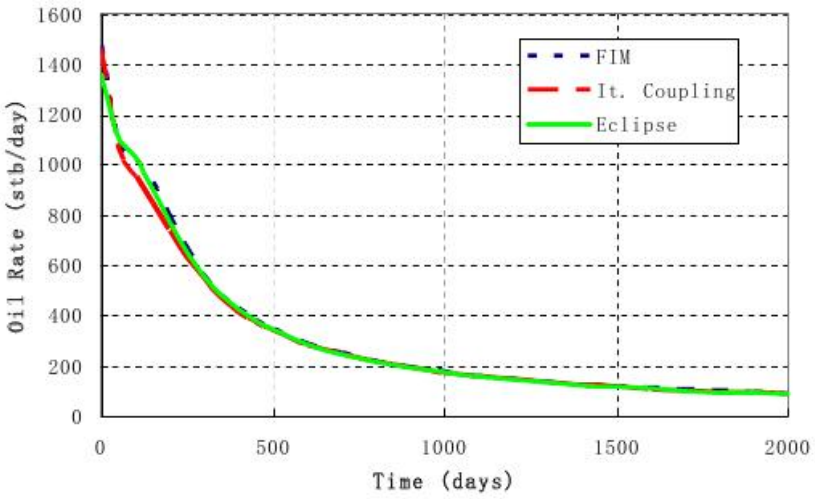

Figure 3.8: Oil production rate in producer 3

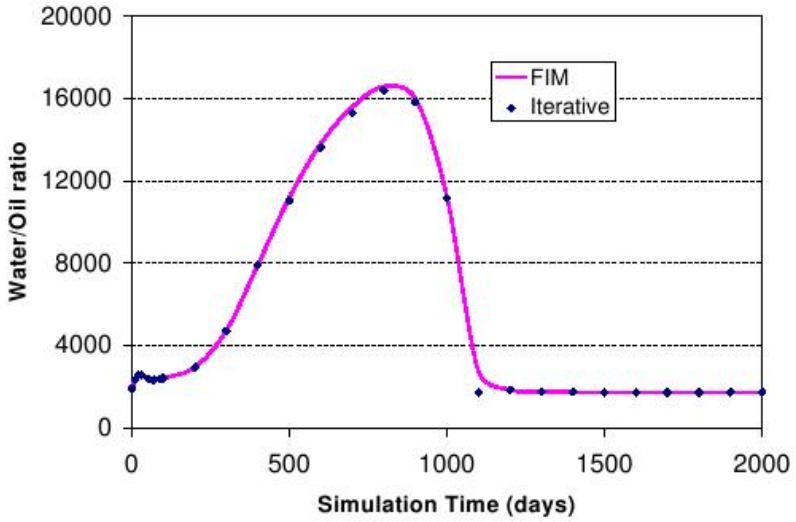

Figure 3.9: Water/oil ratio of producer 3

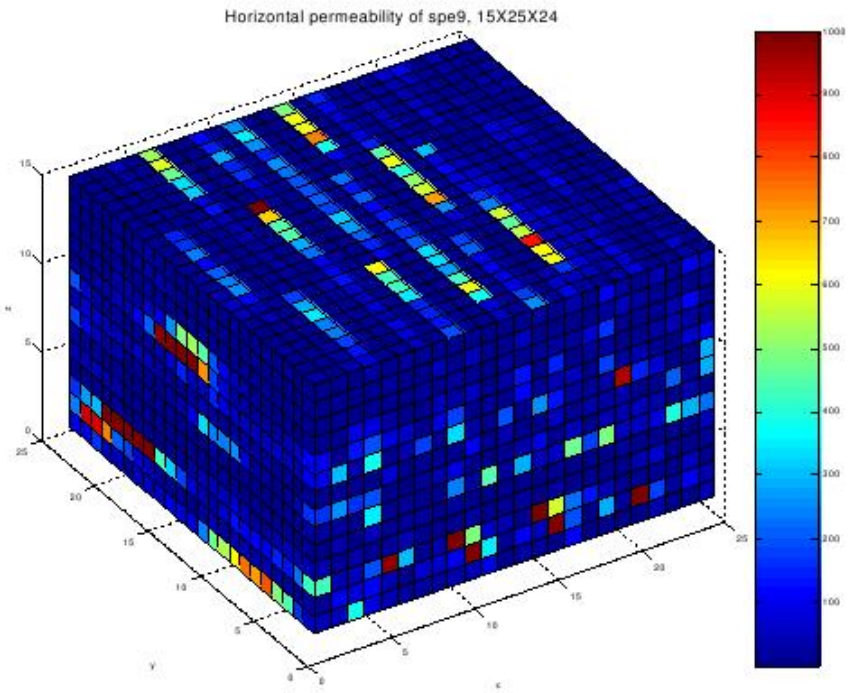

Figure 3.10: Horizontal permeability for SPE 9 reservoir

3.4.3 SPE 9 Problem

This numerical example comes from the ninth SPE comparative project [68]. The reservoir has 9000 grid elements with high heterogeneity of permeability, as shown in Figure 3.10 and 3.11. High permeability zones are distributed in a few layers with poor connection. Other PVT properties and a reservoir description can be found in the references. The reservoir hosts 26 wells, including 1 injector, and all of them are bottom hole pressure specified.

The total simulation time is 900 days, and the maximum time step is set to 5 days. As shown in Table 3.3, with the iterative method, the total CPU time is reduced from 8017s using the FIM to 6114s. The iterative method obtains very good

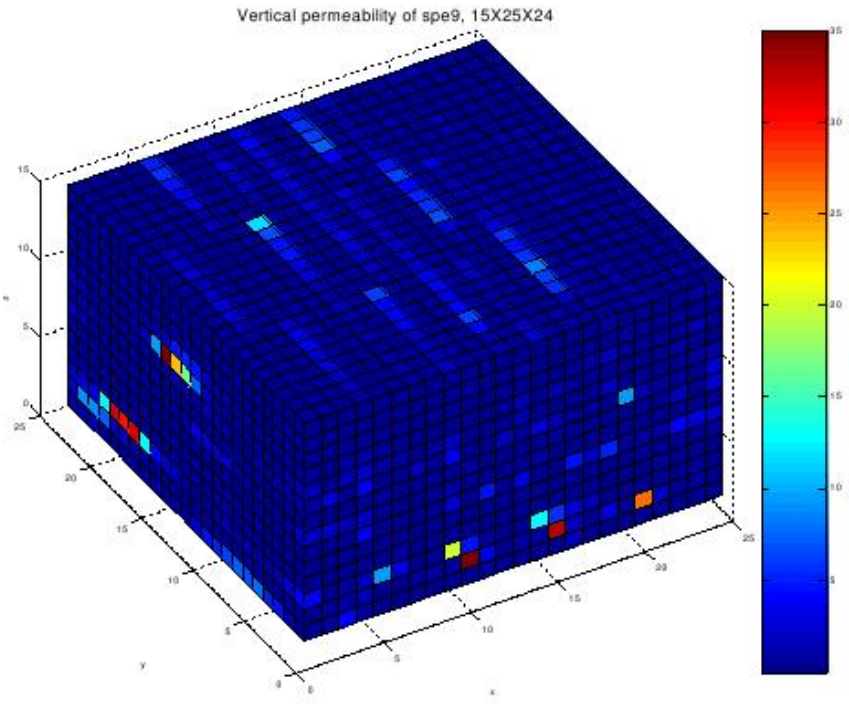

Figure 3.11: Vertical permeability for SPE 9 reservoir

Model	Oil balance	Water balance	Gas balance	CPU time (sec.)
Iterative	1.0003205	0.9992816	1.0006002	6114
FIM	1.0000012	0.9999990	1.0000047	8017

Table 3.3: Timing comparison for SPE 9

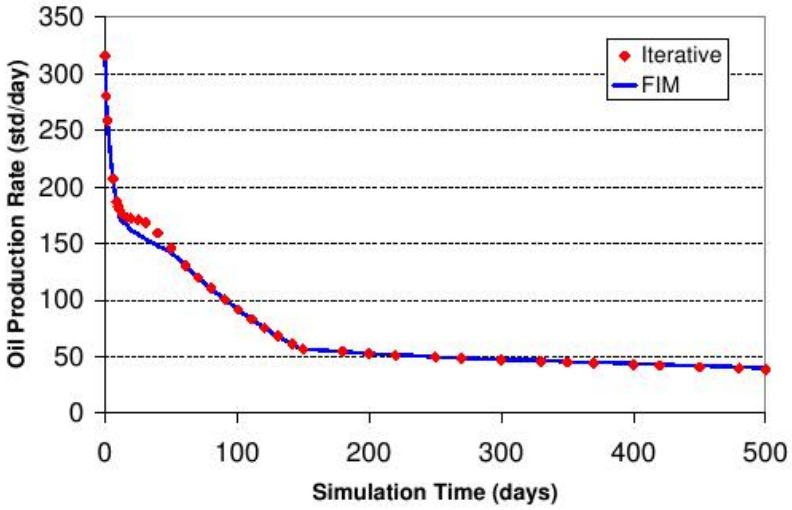

Figure 3.12: Oil rate in producer No.20

matches for both the oil production rate (Figure 3.12) and the gas/oil ratio (Figure 3.13). During the simulation, some time step cutting occurs with the iterative method at the beginning, but it becomes stable after 50 days.

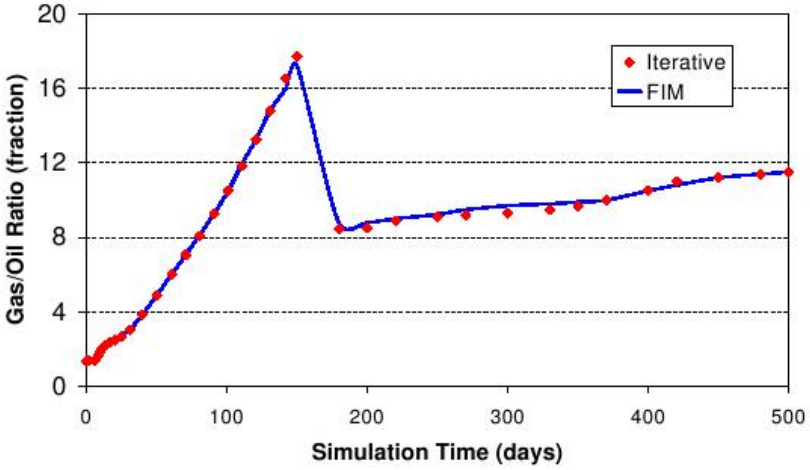

Figure 3.13: Gas/oil ratio in producer No.20

Chapter 4

Linear Solvers for the Iterative Model

Since more than $80-90\%$ of CPU time in a reservoir simulation is spent on solving linear systems, an efficient linear solver is crucial. In this chapter, the discussion focuses on those linear solvers and preconditioners that have been implemented in the IPARS framework, with an emphasis on their application and performance in the iterative model. General theory of linear solvers for fluid flow simulation is not discussed.

4.1 Introduction

4.1.1 Linear Solver

The matrix system of a reservoir simulation is usually sparse, highly nonsymmetric, ill-conditioned, and very large. The *Krylov* subspace algorithm is one of the

best approaches to solve such matrix systems. We have considered three Krylov iterative linear solvers, namely PCG, GMRES and biconjugate gradient stablized (BiCGSTAB). They are very different in their capabilities and suitability to particular linear systems. PCG is very popular for solving symmetric positive definite linear systems. GMRES is the most powerful algorithm for solving general sparse non-symmetric systems. However, with GMRES, the bases of the Krylov space must be stored as the iteration progress, requiring large amounts of computer memory. An orthogonalization truncation or restart can be used to alleviate such large memory requirements [9, 52]. BiCGSTAB involves two preconditioning steps and two matrix-vector products in one iteration. It takes more CPU time than GMRES on an iteration basis but it has low memory requirements. The main linear solver incorporated in the iterative model is GMRES with various preconditioners.

4.1.2 Preconditioner

Since the original matrix for a reservoir flow problem is hard to solve, an intermediate step is proposed to aid on solving the problem. Given the matrix A, a preconditioner M is the approximation of A which makes the preconditioned system easier to solve indirectly using an iterative algorithm. Preconditioners are applied to the solving of both coupled system and decoupled system.

Coupled System

For a coupled system, if M_0^{-1} is a right preconditioner for the matrix A, further preconditioning may be performed on the system $A_0 = M_0^{-1}A$ by means of a preconditioner M_1^{-1}. The concatenation of these two preconditioners is what is called

two-stage preconditioning. Multi-stage preconditioning can be achieved by generalizing this procedure.

Since pressure blocks are much harder to solve than other blocks, the solution of pressure dominates the entire solution process. The two-stage preconditioner first decouples the pressure block from the whole equation system and solves the pressure system itself; then the preconditioner recovers the solution components related to the original coupling or to the non-pressure blocks. In the first stage, the extracted pressure system should preserve the algebraic properties of the original system.

In a fully implicit system, the coupled system has following formula

$$Ax = \begin{pmatrix} A_{pp} & A_{pc} \\ A_{cp} & A_{cc} \end{pmatrix} \begin{pmatrix} x_p \\ x_c \end{pmatrix} = \begin{pmatrix} b_p \\ b_c \end{pmatrix}, \tag{4.1}$$

where $A_{pp} \in \Re^{np \times np}$ represents the pressure block coefficients and $A_{cc} \in \Re^{nc \times nc}$ represents the concentration block coefficient. The other two blocks, A_{pc} and A_{cp}, are the respective coupling coefficients. The block A_{pp} holds a set of coefficients as a parabolic problem, which is diagonal dominant.

Effective extraction of the pressure block from the overall system is realized using *decoupling operators*. The decoupled matrix becomes

$$\tilde{A} = \begin{pmatrix} \tilde{A}_{pp} & \tilde{A}_{pc} \\ \tilde{A}_{cp} & \tilde{A}_{cc} \end{pmatrix}, \tag{4.2}$$

with a residual vector $r = (r_p, r_c)^T$. The two-stage preconditioner then completes the following steps:

- Solve the new pressure system $\tilde{A}_{pp}\delta_p = r_p$;

- Compute the new residual $\hat{r} = r - \tilde{A} \begin{pmatrix} \delta_p \\ 0 \end{pmatrix}$;

- Precondition and correct: $\delta = M^{-1}\hat{r} + \begin{pmatrix} \delta_p \\ 0 \end{pmatrix}$,

where $\delta = (\delta_p, \delta_c)^T$ is the correction obtained after the two stages. In summary, the two-stage preconditioner M_{2s} functions as

$$\delta = M_{2s}^{-1} r = M^{-1} \left[I - \left(A \tilde{-} M \right) \begin{pmatrix} \tilde{A}_{pp}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \right] r. \tag{4.3}$$

Large time steps and small pore volumes in grid blocks may negatively affect the diagonal dominance of the diagonal blocks of A_{pp} and, therefore, compromise the convergence of the iteration solver. This has been proven in our own research as well as in the research of others [75, 76].

Decoupled System

For the iterative model, since the equations system has already been decoupled into a pressure equation and a saturation equation, a two-stage preconditioner is meaningless. Here, a preconditioner is used to provide a better initial guess of the pressure equation.

4.2 SAMG

AMG is a very effective solver for linear systems of fluid flow. Unlike geometric multigrid which is based on coarsening grids, AMG is solely based on the linear

system and operates directly on the matrix that corresponds to a given discretization. There is no need to construct a sequence of nested grids, which is difficult in parallel implementations. Since the explicit construction of a multilevel hierarchy is part of the AMG algorithm, a corresponding solver is easy to integrate into existing simulation packages. In reservoir simulation, AMG has clear advantages over classic one-level solvers, especially for problems of large size or high heterogeneities. The SAMG software library [99, 107], developed by the Fraunhofer Institute for Algorithms and Scientific Computing, is based on the AMG approach, and it has proven to be very flexible and robust. It can be applied to solve certain classes of PDEs on unstructured meshes, both in 2D and 3D.

SAMG has been implemented in IPARS, and it is used both as a linear solver and as a preconditioner. For the FIM, SAMG is either used to solve the coupled system, or it is used as a solver for only the pressure system, which comes with the decoupling of the two-stage preconditioner. For the iterative method, SAMG is either used to solve the pressure system or as a preconditioner for the pressure equation. The performance of IPARS with SAMG is shown later in this chapter.

4.3 HYPRE

HYPRE, developed by the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory, is a software library of high performance preconditioners and solvers for solving large, sparse linear systems of equations on massively parallel computers. It includes parallel multigrid solvers for both structured and unstructured grid problems.

HYPRE contains several families of preconditioner algorithms that focus on

the scalable solution of very large, sparse linear systems. It provides several of the most commonly used Krylov-based iterative methods for use in conjunction with its scalable preconditioners. These methods include those for non-symmetric systems, such as GMRES, and methods for symmetric matrices, such as conjugate gradient. Several other options are also available, such as the semicoarsening multigrid (SMG), fast adaptive composite grid (FAC), Maxwell, Hybird, block Gauss-Seidel (SplitSolve) *et al.*

For easy of use, HYPRE employs a conceptual interface to link and pass the linear system problem from the numerical model into the linear solver. Four interfaces are currently provided by the HYPRE package: the Structured-Grid System Interface (Struct), the Semi-Structured-Grid System Interface (SStruct), the Finite Element Interface (FEI), and the Linear-Algebraic System Interface (IJ). Interfaces are chosen by the user to match specific applications, where more efficient solvers and preconditioners are applied. For the iterative model, the semi-structured interface is used with GMRES as the main solver, and BoomerAMG (parallel Algebraic Multigrid) as the preconditioner for solving the equation system. The performance of this solver package is illustrated in the parallel scalability study of Chapter 5.

For the purpose of applying HYPRE to the IPARS framework in both the FIM and the iterative model, an interface subroutine is designed to obtain the coefficient matrix from IPARS and then pass it to the HYPRE library, which is very similar to the method that integrates other solvers, such as GMRES, with IPARS. A separate subroutine has been built to calculate the reservoir grid distribution among multiple processor. It provides all information required for communication, including global and local grid indexes, neighbouring processors, and exchanging arrays. Since HYPRE conducts its own parallel communications in one step, only the matrix for

Case	Mesh	Reservoir
Case 1	60×220	1 slice of non-fluvial reservoir
Case 2	60×220	1 slice of fluvial reservoir
Case 3	$12 \times 44 \times 17$	Upscaled entire reservoir
Case 4	$15 \times 55 \times 25$	Upscaled upper subdomain
Case 5	$30 \times 110 \times 50$	Upsacled upper subdomain

Table 4.1: Numerical cases for solver study

the subdomain problem is passed to the corresponding processor. In addition, some parameters that are used by HYPRE must be defined by the user, either in the solver subroutine or in the input file. The solver returns information about the linear solution, such as number of iterations, relative and absolute residuals.

4.4 Numerical Example

The implementation of SAMG and HYPRE on parallel environments is discussed later in Chapter 5. This section addresses the application of SAMG on a single processor. Five test cases are established based on different subdomain and upscaled grids of the SPE 10 reservoir. These cases are listed in Table 4.1. As in Section 2.6.2, five wells are defined on each of these reservoirs, with varying depth according to the reservoir specification.

4.4.1 Two-phase Problem

Both the FIM and the iterative model are used to solve the two-phase problem with the following four solver options:

- FIMS-SAMG: use SAMG as the solver for the coupled system with the FIM;

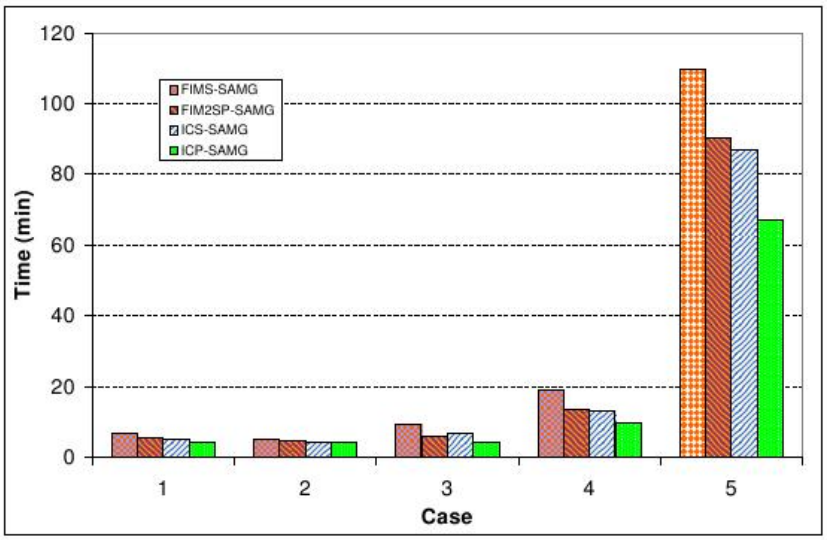

Figure 4.1: SAMG solver with the two-phase model

- FIM2SP-SAMG: use SAMG in the two-stage preconditioner for solving the pressure system;
- ICS-SAMG: use SAMG as the solver for the pressure system in the iterative model;
- ICP-SAMG: use SAMG as the preconditioner for the pressure system in the iterative model.

All of the simulations in the 5 cases above are carried out for 2000 days, except for case 5, which is simulated for 500 days. Results are shown in Figure 4.1. For the FIM, two-stage preconditioning offers slightly better performance when compared to directly solving the coupled linear system with SAMG as the solver. The iterative coupled models are much faster than the FIM. The solver option where GMRES is preconditioned with SAMG produces the best performance.

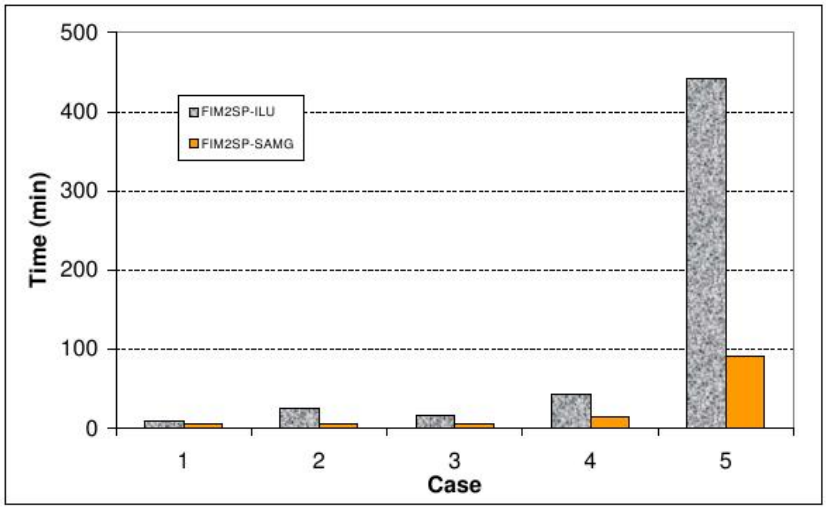

Figure 4.2: Comparison between preconditioning with SAMG and preconditioning with multilevel ILU

Another comparison is made between the solver options of GMRES preconditioned with SAMG and GMRES preconditioned with multilevel ILU. It turns out that SAMG holds clear advantages over multilevel ILU (Figure 4.2).

4.4.2 Black Oil Problem

The black oil model is applied to simulate all of the 5 cases in the previous section. The reservoirs remain the same, and PVT properties for the black oil system are modified from SPE 9. The results are shown in Figure 4.3. SAMG is very efficient as a basic linear solver, and it is even more efficient as a preconditioner. This is also true for both the FIM and the iterative model.

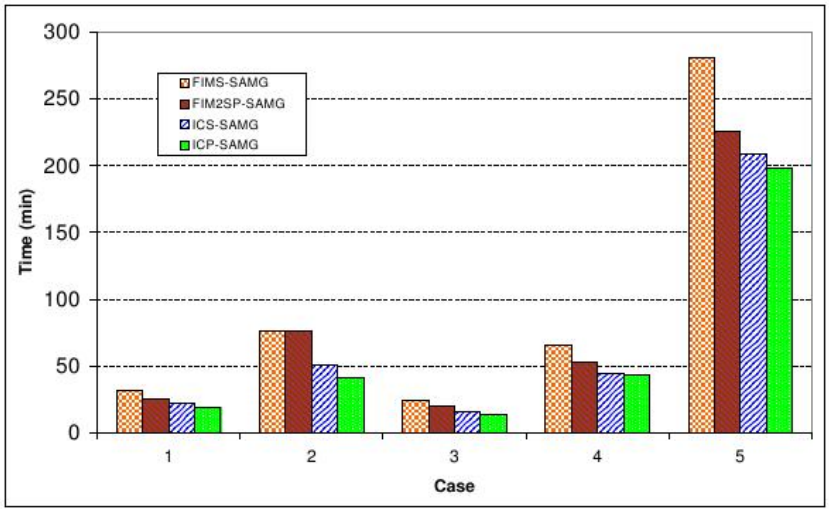

Figure 4.3: SAMG solver for the black oil model

Chapter 5

Parallel Implementation

The ability to run on parallel clusters has become a standard functionality of the new generation of reservoir simulators. Since different simulators use different domain decomposition methodologies and different parallel language libraries, parallel scalability varies from low to high across a broad range. This chapter discusses the parallel implementation of the iterative model and its performance in multiple processor environments. The use of MPI in IPARS to realize parallel communication and a pre-process strategy for optimal parallel performance are discussed as well. Incorporation of the iterative model with different high performance linear solvers in HPC environments is also compared.

5.1 Introduction

5.1.1 Parallel Computation Efficiency

Parallel efficiency is used to represent the scalability of a program running with multiple processors. For a given problem, the simulation time of a perfectly par-

allelized program is linearly decreasing with the increasing number of processors. Unfortunately, some program parts are non-parallel, and several other issues exist to slow down parallel computation.

All parallel programs contain a parallel section f_p and a serial section f_s. The work of the parallel section is distributed among multiple processors, while the work of the serial section is done on the master processor alone. The portion of serial section limits the parallel effectiveness. Amdahl's law states this limitation formally. It places a strict limit on the speedup that can be realised by using multiple processors.

The effect of multiple processors on running time is given by

$$t_n = (f_p/N + f_s) \times t_1, \tag{5.1}$$

and the effect of multiple processors on speed up is given by

$$S = \frac{1}{f_s + f_p/N}, \tag{5.2}$$

where N is the number of processors and t_1 is the running time on a single processor. Figure 5.1 shows Amdahl's law for an increasing number of processors. With different parallel portions, the three programs perform very differently. The affects of the serial code accumulate dramatically during multiple processor simulation. For example, the program that has 99% parallel code only achieves 50% parallel efficiency with 100 processors. Amdahl's law gives an estimation of parallel efficiency under ideal conditions. The practical situation is even worse than the prediciton of Amdahl's law. Several other considerations, such as load balancing and communications, also degrade parallel efficiency.

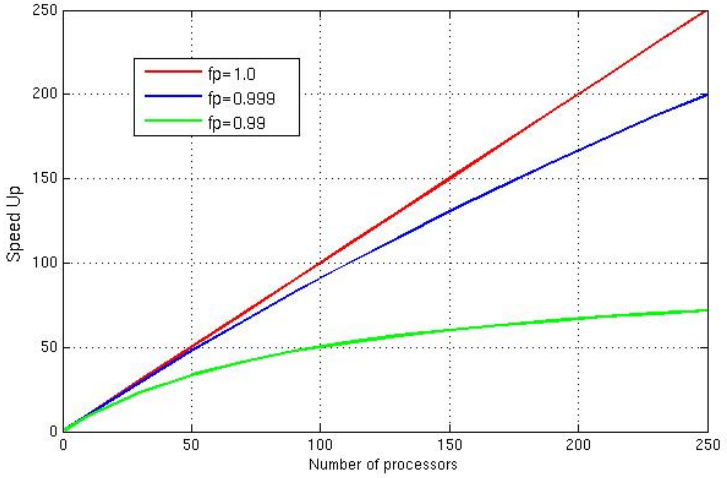

Figure 5.1: Illustration of Amdahl's law

Load balance indicates how well work has been assigned across multiple processors. In reservoir simulation, two main methods are used. One is grid-based load balancing, and the other is reservoir-based load balancing. Grid-based load balancing tries to assign all grids as evenly as possible among all processors. Reservoir-based load balancing considers the heterogeneity of the reservoir and then tries to distribute computation work evenly across all processors. In IPARS, grid-based load balancing is used in simulations with single physical model; reservoir-based load balancing is used in simulations with multiple physical models [77, 78].

Communication with neighbouring processors is necessary during parallel computation in order to solve the problem globally. Information exchanging and updating occur within every time step. The bigger the number of processors, the more communication happens. Figure 5.2 shows the total grid elements involved in parallel communication in one time step for a given reservoir with $56,100$ elements and a 7-point stencil algorithm. With the number of processors changing from 2 to 64, the

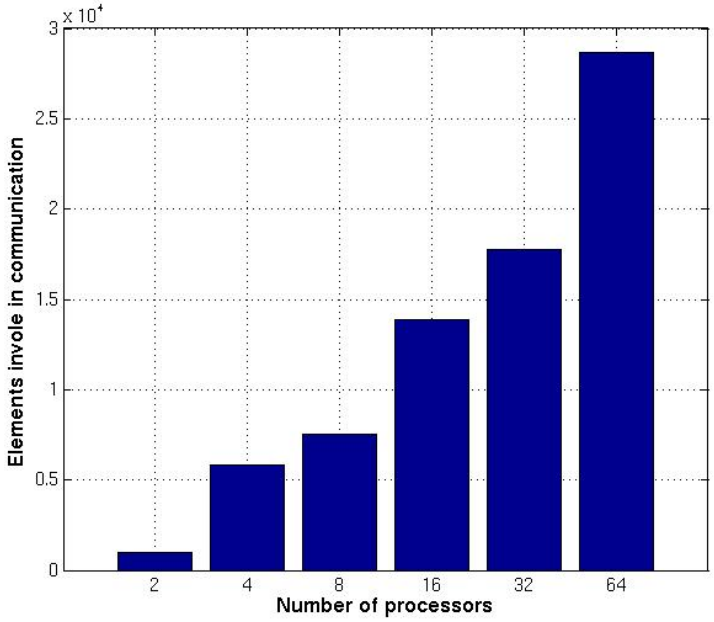

Figure 5.2: Communication levels during parallel computation

number of elements that involved in communication increases from $1,020$ to $28,645$. Note that these elements are involved in local computation too. Additional computation time is required to cover this overlap, which slows down the entire parallel computation procedure.

5.1.2 Parallel Environments

The most popular parallel computers can be classified into either *shared memory system* or *distributed memory system* according to their memory model. For shared memory system, all processors have access to the memory pool with a single address space. Bus and crossbar are used as the bridge to connect the processors with the memory. For distributed memory system, each processor has it's *local* memory,

which is connected with various topological interconnects. Exchanging data between processors is realized by message passing.

Accordingly, there are two programing languages in use for parallel computation. The first is MPI for distributed memory system, and the second is OpenMP for shared memory system. The use of different languages may lead to different performance. On a shared memory system, the communication work is tremendously decreased due to sharing data set in the public cache. Additionally, OpenMP is easy to implement; just a little work is required to convert a serial code into parallel code with OpenMP. However, the shared memory system has difficulties supporting large number of processors, blocking it from applications to huge reservoirs with millions of elements. Distributed memory system is highly scalable with large number of processors and is used as the main library on huge clusters; but it is much harder to implement than OpenMP. IPARS is using MPI to bridge the parallel communication.

5.1.3 Parallel Efficiency-related Issues

Parallel efficiency is a performance metric that is tied to a combination of various issues. In parallel reservoir simulation, with respect to the reservoir, such issues include the reservoir dimension, mesh size, time step, included files *et al.*; with respect to the architecture, such issues include the communication method, CPU speed, memory size and network bandwidth *et al.*; with respect to a program, such issues include the domain decomposition methodology and solver parallelism, as well as the parallel portion of the code. In evaluating the parallel efficiency of a simulator, all of these factors should be considered.

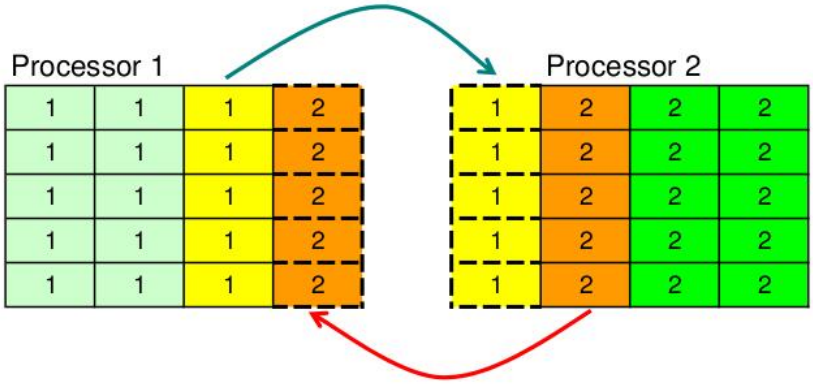

Figure 5.3: Communication through ghost cells

5.2 Parallel Implementation of IPARS

IPARS is well designed for parallel computation. Message exchanging is realized using the MPI library. Communications between neighboring processors are conducted through a ghost cell, as shown in Figure 5.3. On processor 1, the layer close to the neighboring processor is designated as the ghost layer, and it is also designated as the boundary layer on processor 2. Similarly, the ghost layer on processor 2 is also designated as the boundary on processor 1. During parallel computation, values on the ghost layers are exchanged between the neighboring processors. With more processors and irregular grids, the situation is more complicated but the same strategy is followed.

For simulations of individual physical model, IPARS uses grid-based load balancing to distribute the reservoir grid blocks across multiple processors, as in the following steps:

- Calculate the total number of grid blocks for the reservoir.

- Assign a mapping index to each vertical column of the grid blocks (all grid blocks in the vertical column (x, y) will be allocated to the same processor).
- Calculate the average number of grid columns for a given number of processors.
- Allocate a rectangular region (or a region close to rectangular) of grid columns to each processor one by one.

This method tries to distribute the average number of grid columns as equally as possible across all available processors. The topology of the load balancing for 2D and 3D reservoir is shown in Table 5.1 and Figure 5.4. In the 2D case, a reservoir with a 5×6 mesh is distributed across 8 processors. In the 3D case, a reservoir with a grid mesh of $30 \times 110 \times 17$ is allocated across 20 processors. Each color in Figure 5.5 represents a single processor. All grid blocks within one vertical column are allocated to the same processor.

This grid-based load balancing method has proven to be very efficient, according to simulation results. For a simulation using 20 processors for a given problem, the CPU time used throughout of the simulation on each processor was recorded and sorted. The maximum and minimum values were very close, indicating that every processor made a similar contribution to the simulation. The same results have been achieved with different numbers of processors as shown in Figure 5.5.

5.3 Numerical Examples

In this section, several numerical examples are presented to illustrate the parallel scalability and efficiency of the iterative model on HPC. In computer science, parallel scalability is investigated by changing both the problem size and the number of

I ⟶						
J ↓	0	0	3	3	7	7
	0	1	3	3	7	7
	1	1	4	4	6	6
	2	2	4	4	6	6
	2	2	5	5	5	5

Table 5.1: Processor mapping on grid blocks in 2D

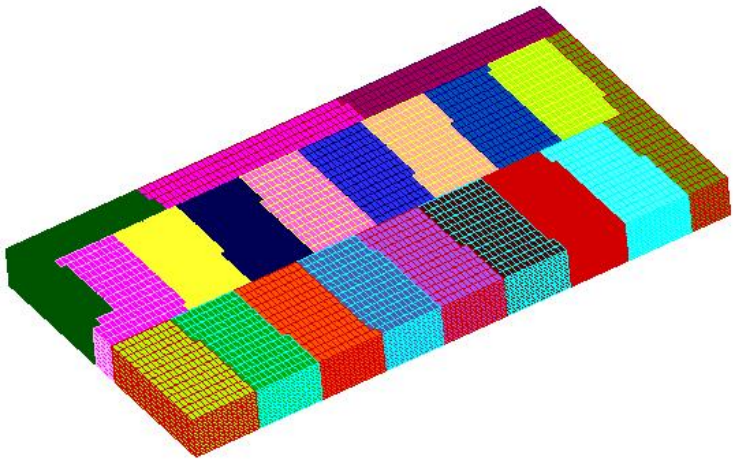

Figure 5.4: Processor mapping on grid blocks in 3D

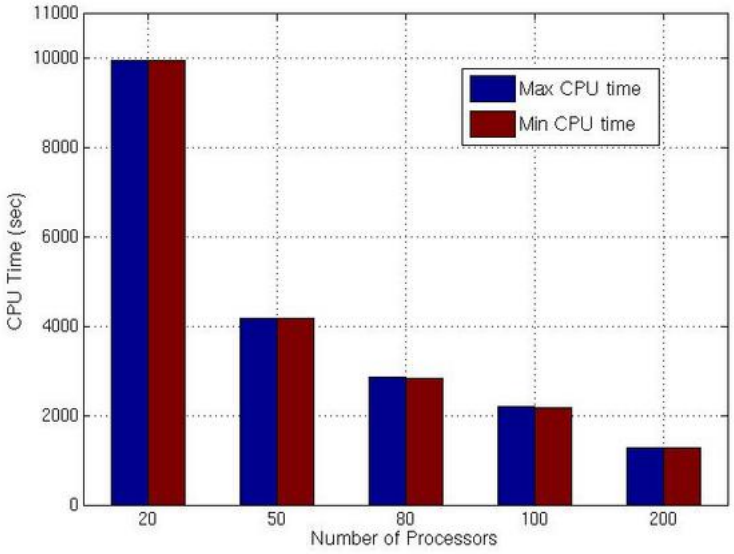

Figure 5.5: Load balancing across multiple processors

processors. The primary goal is to maintain the work load on each processor at a constant level. In different runs, one processor solves subdomain problems of similar size. This approach requires a reasonable estimation of the load balance of each processor and it is usually hard to realize. For the parallel scalability study in this dissertation, fixed reservoir problems running on varying numbers of processors were used. An optimal number of processors for a given problem is discussed later.

5.3.1 HPC Architecture

Two very different HPC clusters are used in this study. One is *Lonestar* from the Texas Advanced Computing Center [108], and the other is *Blue Gene* from the IBM Rochester Blue Gene Center[63]. The Lonestar cluster is one of the largest academic computational resources in US. This dual-core Linux cluster has 5200 processors and

	Lonestar	Blue Gene
Processor	Dual-core Xeon 5100, 22.6GHz	700MHz PowerPC 440
Architecture	64-bit	32-bit
Memory	2GB per processor	512M per processor
Cache	L1=32KB, L2=4MB	L1=32KB, L2=2KB, L3=4MB
Network	InfiniBand Switch, 1GB P2P Bandwidth	3D torus, 175 MBps in each direction
Nodes	1300 nodes, 5200 processors	1024 nodes, 2048 processors per rack
Peak performance	10.64 GFLOPS	2.8 GFLOPS

Table 5.2: Architecture of HPC resources

10.4 TB of total memory, with a peak performance of 55 TFLOPS. The Blue Gene cluster is one of the largest clusters in the world with IBM technology. The main specifications of these two HPC are listed in Table 5.2.

5.3.2 Linear Solver

The following are three linear solver options that are used in the HPC simulations discussed in this dissertation:

- Option 1: GMRES preconditioned with multilevel ILU
- Option 2: GMRES preconditioned with SAMG
- Option 3: HYPRE, GMRES preconditioned with BoomerAMG

The parameters used for SAMG were introduced in Chapter 4. The parameters used in HYPRE are sorted in Table 5.3. A set of numerical examples have been designed to compare the performance of these linear solvers for HPC reservoir

	Parameter	Value
GMRES	Max Num. of Itns	100
	Tolerance	1.0^{-6}
	Max size of Krylov space	5
BoomerAMG	Coarsening	Falgout
	Max Num. of Itns	1
	Tolerance	0.0
	Threshold	0.5

Table 5.3: Parameters used for HYPRE

Case	Grids	Model	Num. of Proc.
Case 1	56,100	oil/water	32
Case 2	224,400	oil/water	100
Case 3	1,112,000	oil/water	150
Case 4	40,050	black oil	150

Table 5.4: Numerical cases for the linear solver on HPC

simulation, as in Table 5.4. The first three cases are modified from SPE 10 and run with the two-phase oil/water model; the fourth case is modified from SPE 9 and run with the black oil model. All of these problems are run on the Lonestar machine with different numbers of processors respectively. The simulation results are shown in Figure 5.6.

For all three two-phase problems, GMRES preconditioned with ILU takes the longest time to complete the simulation, which is about 2 times slower than the other two solving options. HYPRE is slightly faster than option 2 with SAMG, which becomes clearer with larger number of processors. For the black oil problem, options 2 and 3 take a similar amount of time to solve the problem. Again option 1 takes about 3 times as much time as the other two options. Since the parallel package of SAMG has only been verified with less than 256 processors, HYPRE will be used as the default solver option in the following scalability study with large

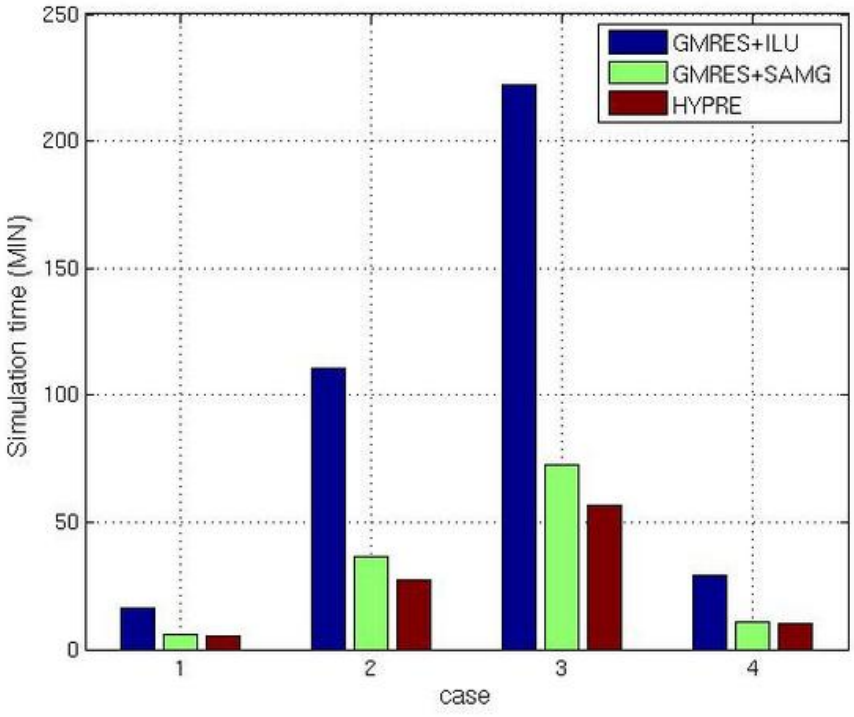

Figure 5.6: Comparison of HPC linear solvers

numbers of processors.

5.3.3 Lonestar

Fine Grid SPE 10

The fine grid SPE 10 problem has a grid size of $220 \times 120 \times 85 = 1,122,000$. The reservoir dimensions are $2200ft$ long, $1200ft$ wide, and $170ft$ thick. One water injection well is located at the center of the reservoir, and four oil production wells are located at the four corners. All of these wells are bottom hole pressure specified. This problem has been run with a series of processors from 10 to 200. Parallel

efficiency is illustrated in Figure 5.7.

An assumption has been made for the parallel scalability study. Since most problems are too big to be finished with a small number of processors within a limited time, it is assumed that the iterative model is perfectly scalable with a small number of processors. For every numerical example, 100% parallel efficiency is assumed in the simulation with the least number of processors. For example, we assume that 10 times speed up has been achieved for this SPE 10 problem with a fine grid and 10 processors. Simulation times for fewer processors is back calculated according to this theory. The same assumption has been made for all of the numerical examples below. This assumption has been proven appropriate according to several case studies with small problems by the author and in previous parallel simulations.

For those simulations with no more than 50 processors, high parallel efficiency close to 100% has been achieved. Super efficiency has been achieved for a simulation with 80 processors. As anticipated, parallel efficiency decreases with an increasing number of processors. Even though, with 150 processors, the iterative model achieves 139 times speed up, which is 93% parallel efficiency. Several factors may contribute to the super efficiency, such as a good match between the problem size and the number of processors, perfect load balancing, efficient parallel linear solver *et al.*

AMV Problem

This example is modified from a real reservoir, and all reservoir properties are derived from a geological map. The reservoir has dimensions of $111,952.94ft$ length, $130,806ft$ width, and $2,400ft$ thickness. The grid mesh is $292 \times 275 \times 63 =$

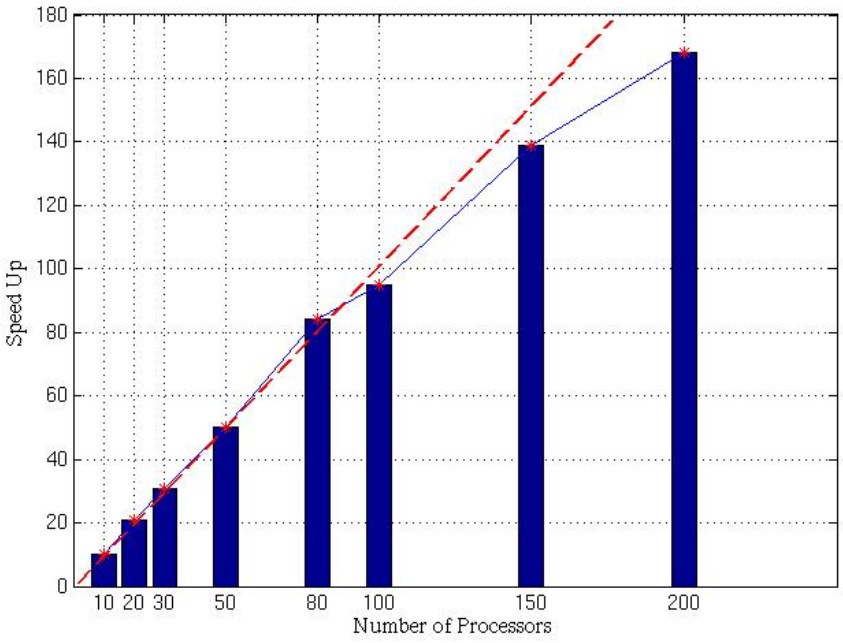

Figure 5.7: Parallel scalability on Lonestar with SPE 10

$5,058,900$. This is a heavy oil reservoir with $API = 8.5$; the oil viscosity has a highest value of $3000cp$. The average temperature of the reservoir is $137F$. Average porosity is about 30%, and permeability varies from less than 10md to 10D. In the real reservoir, more than one hundred wells are located on the reservoir producing a production rate of about $150,000BD$. In this simulation, 16 wells, consisting of one injector and 15 producers, are used to represent this production rate. All of these wells are bottom hole pressure specified. The total simulation time is 2000 days. The speedup achieved with various numbers of processor is illustrated in Figure 5.8. 50 times speedup has been achieved in a simulation with 50 processors; 181 times speedup has been achieved in a simulation with 250 processors, which is more than 72% parallel efficiency.

Refined AMV Problem

This example is modified from the original AMV problem by doubling all three dimensions. The mesh size of the model becomes $584 \times 550 \times 126 = 40,471,200$. The porosity and permeability values are interpolated from the original mesh. Other information is copied from the previous example. This problem has been run with series processor sets, and a 326 times speedup was achieved in a simulation with 500 processors (Figure 5.9).

The distribution of the total CPU time used in these simulations is displayed in Figure 5.10. The linear solver occupied most of the CPU time in all simulations, requiring up to 92% with 50 processors. Parallel communication, initialization, and other sections occupy the rest of the CPU time. It was shown in section 5.1.1 that as the number of processors increases, more time is spent on parallel communication. Figure 5.10 shows the same phenomena as the portion of communication increases

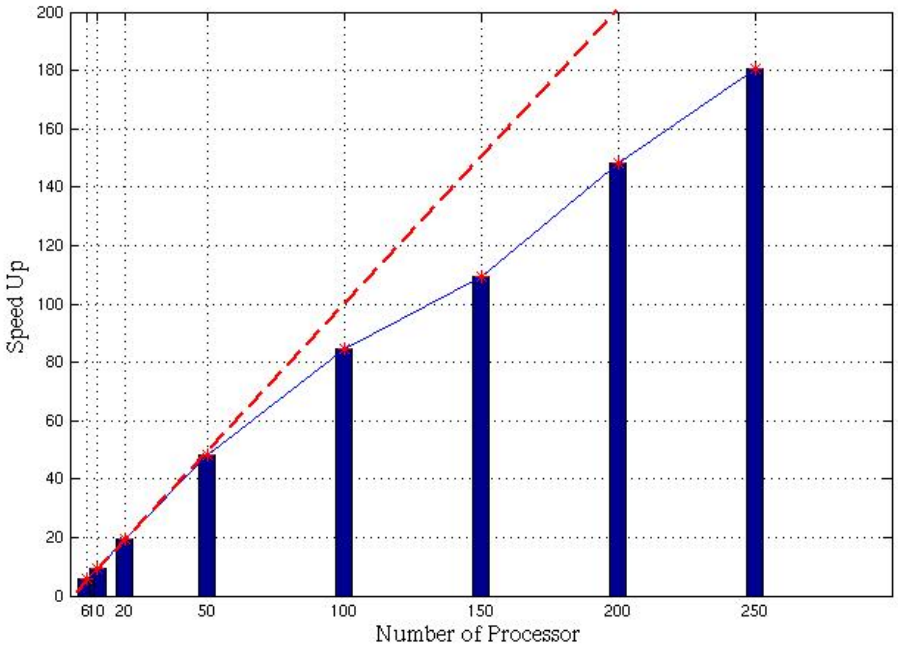

Figure 5.8: Parallel scalability on Lonestar with AMV

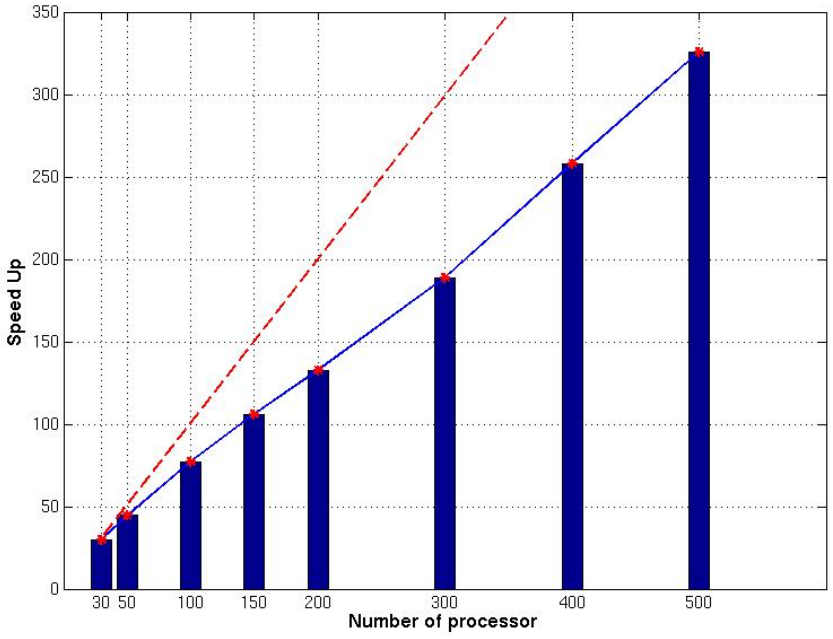

Figure 5.9: Parallel scalability on Lonestar with refined grid AMV

from 2% to 15%.

Since part of the initialization is in serial code, its increasing share of CPU time is mainly due to the decrease in the total simulation time. It also means that it is possible to improve the parallel efficiency of the iterative model by modifying initialization. Figure 5.11 illustrates the new parallel efficiency seen in the iterative method after subtracting the initialization part from the total CPU time in Figure 5.9. With 500 processors, 350 times speed up has been achieved, which is a parallel efficiency of 70% with a 5% improvement over previous results.

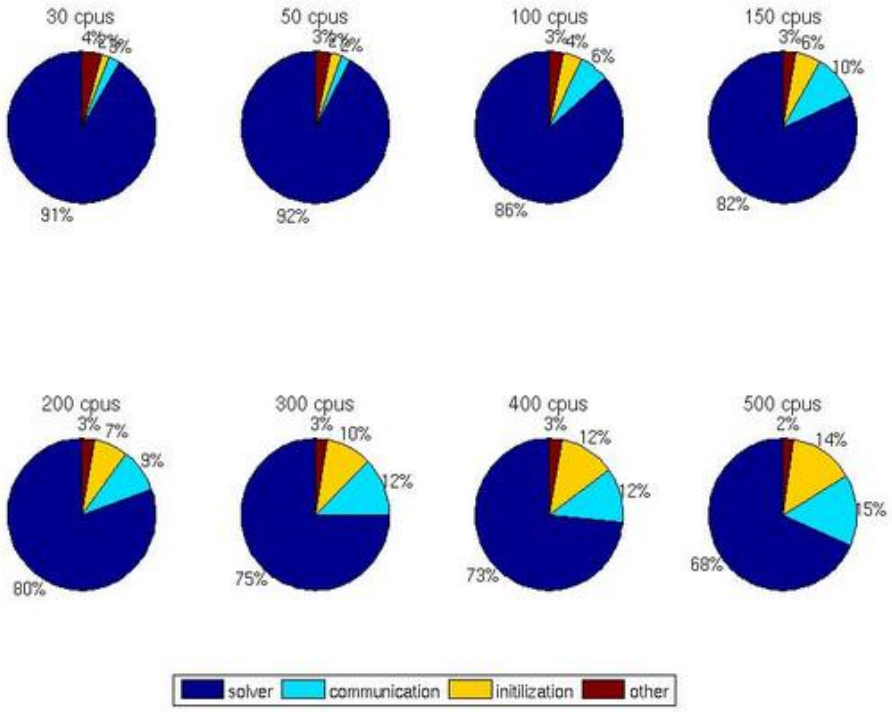

Figure 5.10: CPU time distribution during simulation on Lonestar

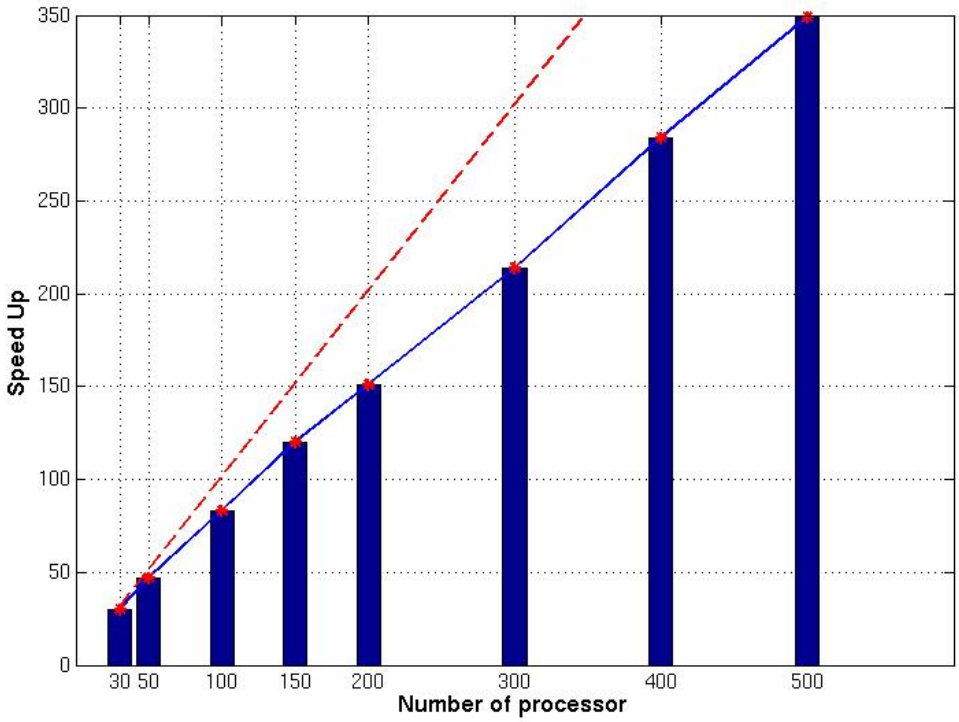

Figure 5.11: Improved parallel scalability on Lonestar with refined AMV

5.3.4 Blue Gene

Upscaled SPE 10

This is an upscaled case of the SPE 10 problem with a grid mesh of $110 \times 30 \times \times 17 = 56,100$. The heterogeneity of the reservoir is reflected in the porosity and permeability seen in Figures 5.12 and 5.13. The upper subdomain of the reservoir is more permeable with high permeability channels. The lower subdomain is less heterogeneous with average low permeability.

For simulations with no more than 100 processors, high parallel efficiency above 90% has been achieved. As the number of processors increases, parallel efficiency decreases gradually, down to around 71% efficiency with 250 processors (Figure 5.14).

Fine Grid SPE 10

In this case, the same problem presented in Section 5.3.3 is solved on Blue Gene, using up to 2048 processors. Figure 5.15 shows that, with 2048 processors, the problem is solved with a 1452 times speed up. The parallel efficiency of this particular simulation is more than 70%. Again the CPU time distribution is shown in Figure 5.16. The parallel communication time increases significantly with large number of processors, which is the main factor to slow down the parallel efficiency.

Again, if subtracting the initialization part from the total simulation time used in Figure 5.15, more than 75% parallel efficiency has been achieved in Figure 5.17. The iterative model offers very impressive scalability with huge numbers of processors.

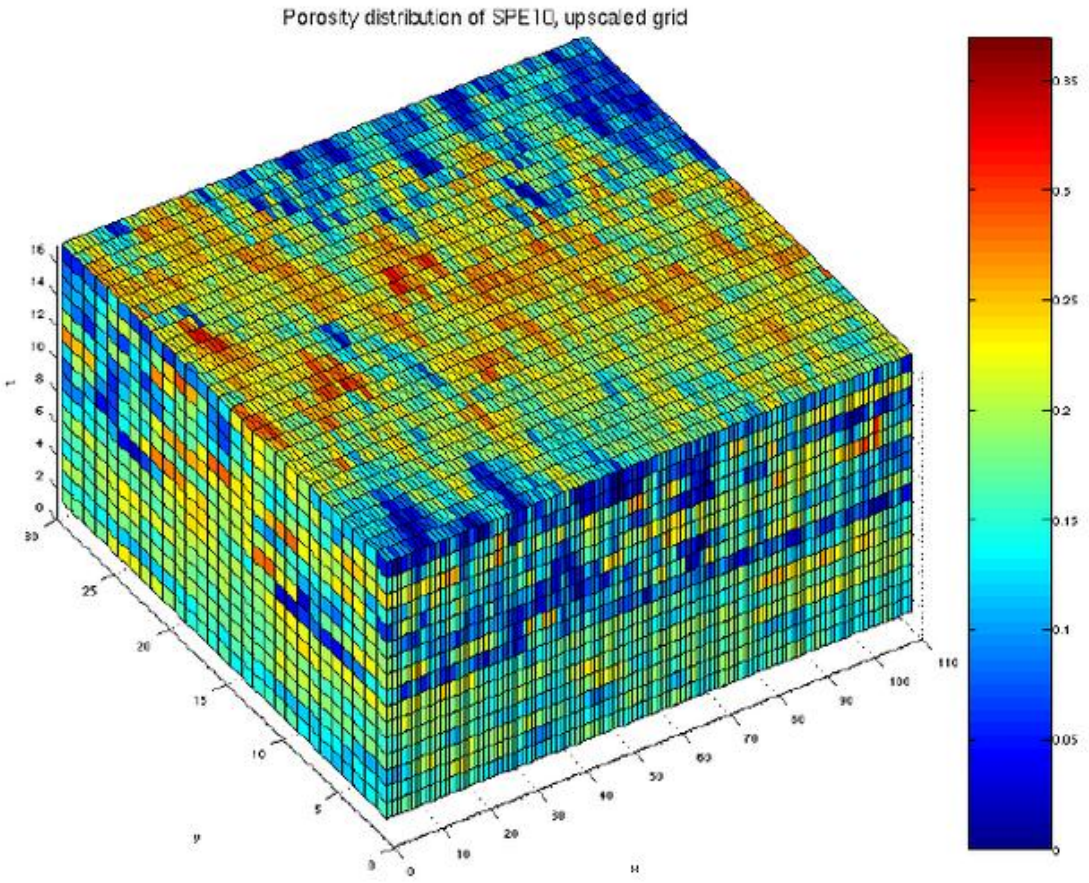

Figure 5.12: Porosity of upscaled SPE 10

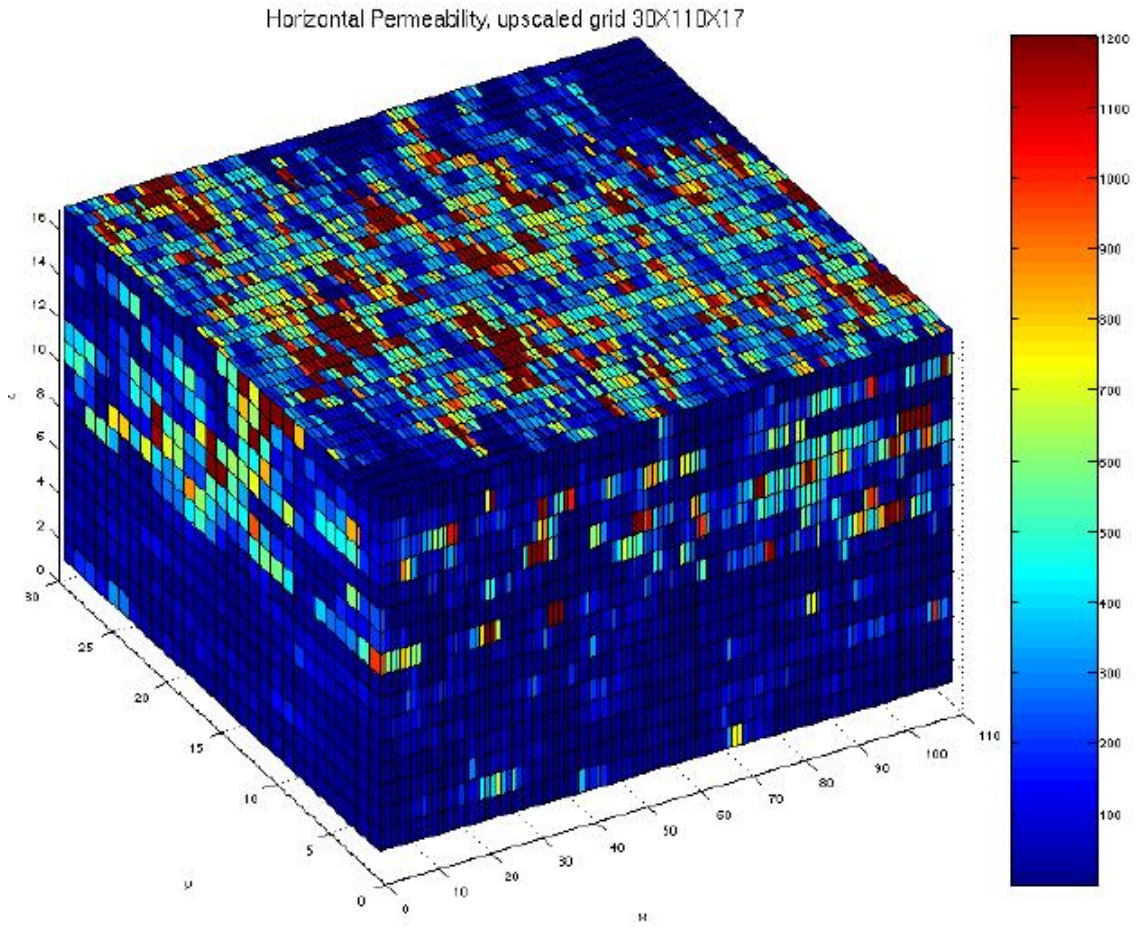

Figure 5.13: Horizontal permeability of upscaled SPE 10

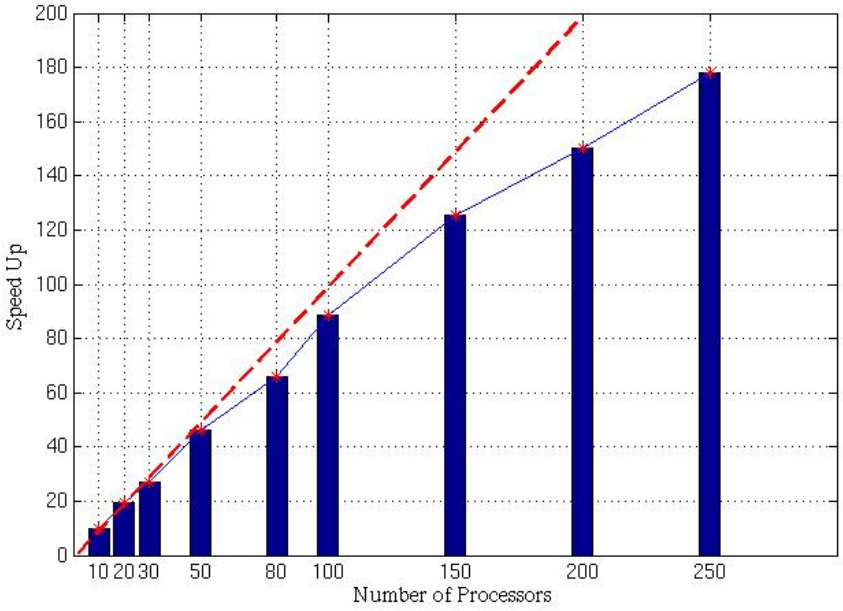

Figure 5.14: Parallel scalability on Blue Gene with upscaled SPE 10

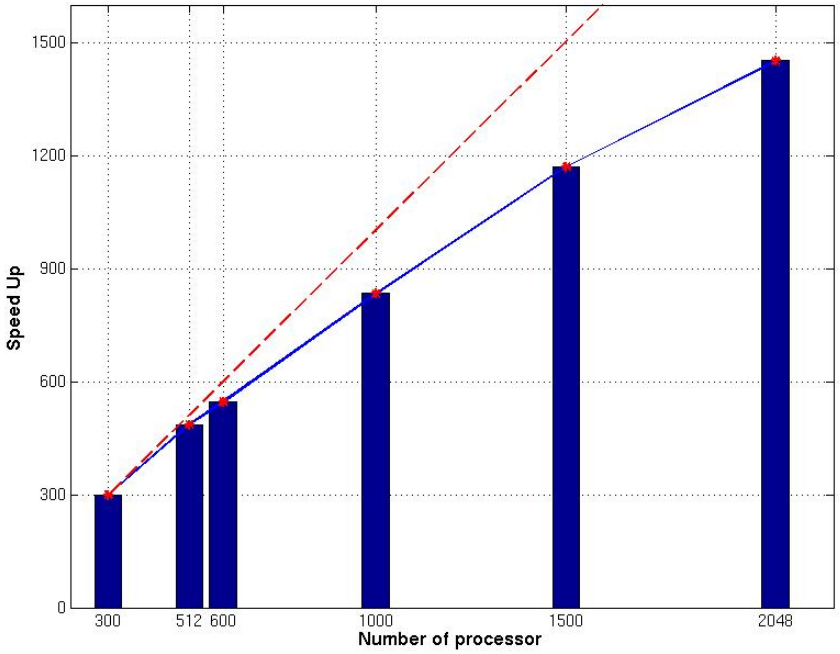

Figure 5.15: Parallel scalability on Blue Gene with fine grid SPE 10

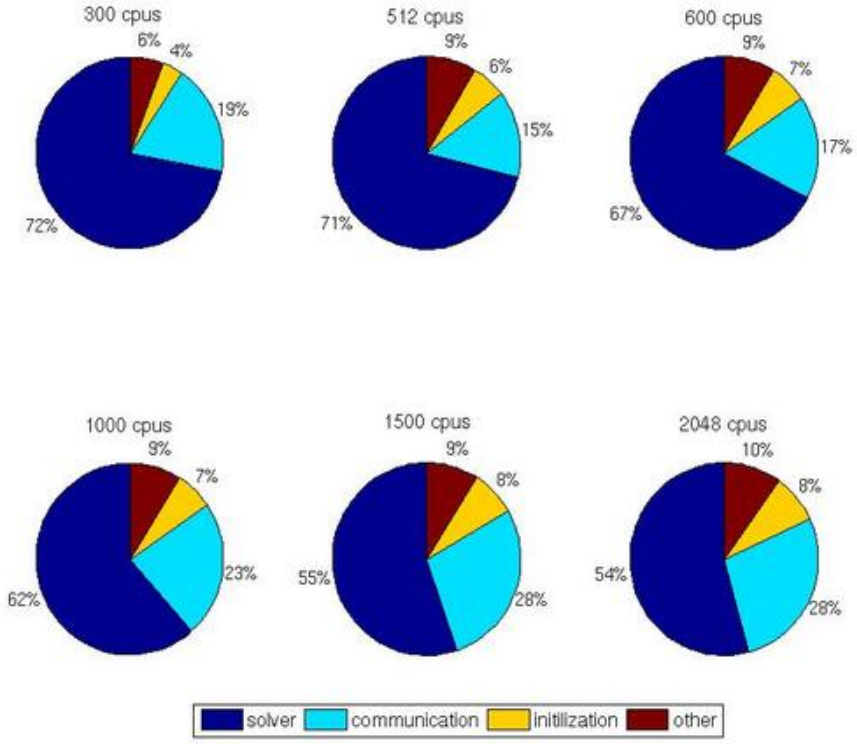

Figure 5.16: CPU time distribution during simulation on Blue Gene

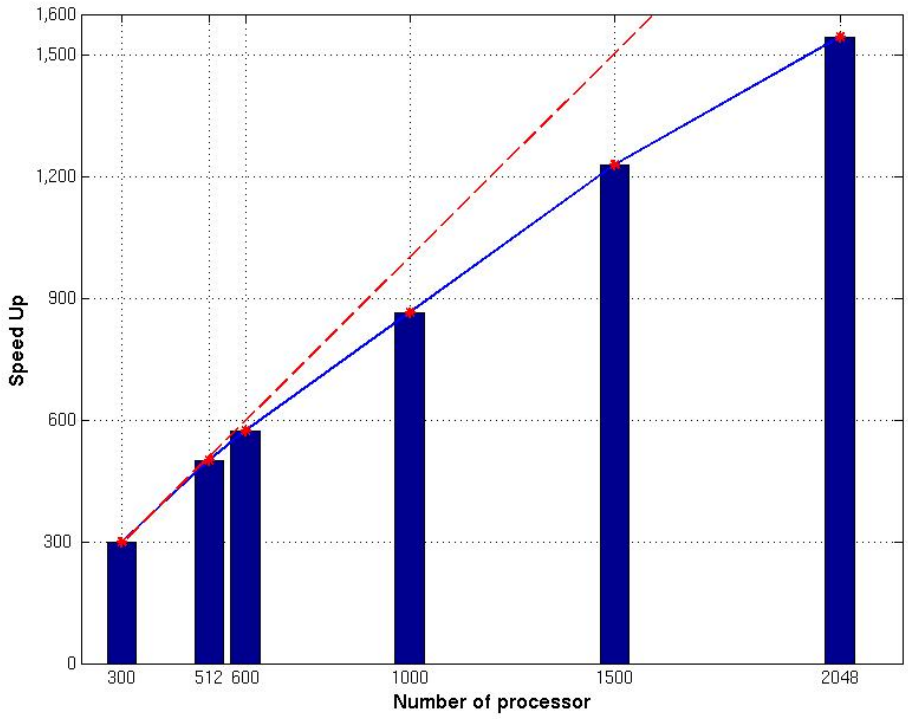

Figure 5.17: Improved parallel scalability on Blue Gene with fine grid SPE 10

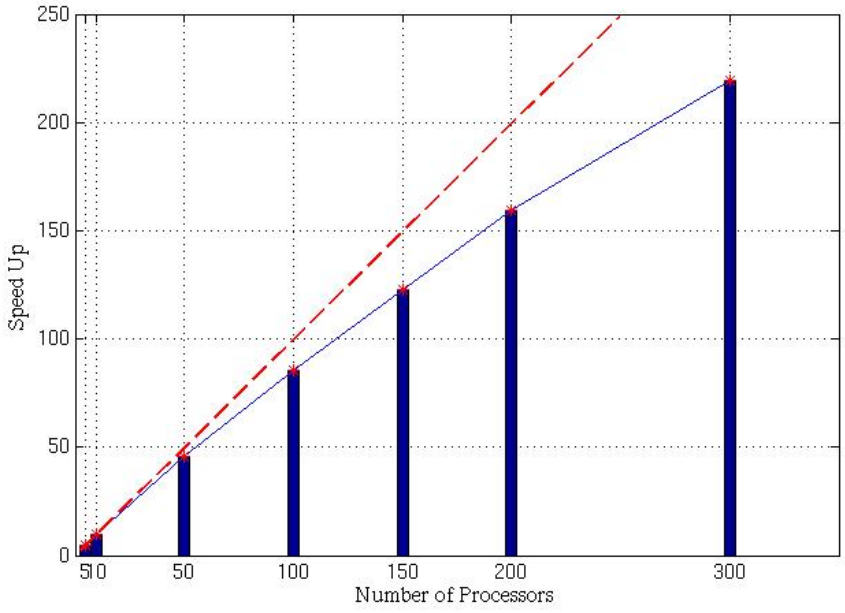

Figure 5.18: Parallel scalability on Blue Gene with SPE 9

SPE 9

This example has the same reservoir as the one discussed in section 3.4.3. This simulation achieves a 223 times speedup with 300 processors, resulting in 74% parallel efficiency (Figure 5.18). It should be noted that the black oil model performs as well as the two-phase model in parallel scalability.

5.4 Discussion

The iterative model shows very good parallel scalability in previous numerical examples, on both Lonestar and Blue Gene with very different architecture. Since most of the CPU time is spent on linear solvers and parallel communications, there

are two ways to improve parallel efficiency correspondingly. A highly scalable linear solver can be used to shorten the linear solve time, and a faster machine can be used to shorten the parallel communication time.

5.4.1 Running with Virtual Memory

Most of the current HPC clusters have dual-core or quad-core technology, with 2 and 4 processors sharing local memory on a computer node respectively. If all processors are used in the simulation, they will share the local memory evenly. For example on a dual-core node with 8GB memory, each processor has a 4GB memory by its own. In case larger memory is required for use, all of the 8GB memory can be assigned to one processor, which means only one of two processors are used during simulation. This virtual memory technology provides flexible applications of HPC considering the heterogeneity of different reservoirs.

However, it brings a new problem to the parallel scalability study in case of using this virtual mode memory. Several problems have been simulated to investigate its affects on parallel efficiency. In each case, two processor sets (1 and 2) are used to solve the same problem. These two sets have the same total number of processors. In set 1, all of the processors come from different nodes; in set 2, some of the processors come from the same nodes. All of the cases are run on parallel clusters with dual-core technology. The results can be summarized as following:

- For small reservoir with limited grid elements, no different observed between processor set 1 and 2;
- For bigger reservoirs that require large memory, processor set 2 is slower than processor set 1;

- Huge reservoirs with millions of grid elements can not run with processor 2 due to its smaller memory limit;
- The following argument has not been observed: processor set 2 could be faster than processor set 1 because part of the communication is conducted locally.

5.4.2 Pre-process

Since parallel efficiency of the reservoir model relates to several issues especially the problem size, a pre-process strategy is adopted in the iterative model for the purpose of best parallel performance. For a given problem, the following issues are considered before starting the simulation.

- Problem analysis: reservoir dimension, mesh size, structure, heterogeneity, time step, included files, wells;
- Numerical model analysis: number of unknowns, explicitness, functionality, memory management, fraction of parallel code;
- HPC architecture: 64bit/32bit technology, CPU speed, memory, network bandwidth, cache, peak performance;
- Linear solver: solver, preconditioner, convergence control, parameters setting, parallel implementation;
- Communication estimation: elements at the interface, messages need to be passed, message size, language interface.

All of above estimations will be put into a size file, which will be integrated into the executable during program compilation. This executable will be used to

Type	Problem size	Num. of Proc.
two-phase	51600	32
	224400	100
	1122000	160
	5058900	280
three-phase	9000	45
	51600	190
	224400	460

Table 5.5: Optimal number of processors on Lonestar

simulate the given problem. In this way, the particular problem is solved with the corresponding executable designed for optimal performance.

5.4.3 Optimal Number of Processors

As discussed before, parallel efficiency is problem dependent. For a given problem, the best performance could be achieved only with a certain number of processors. In Table 5.5, the optimal number of processors for corresponding problems are given based on the simulations with the iterative method on Lonestar. If more processors are used than these given numbers, parallel efficiency will decrease. The estimation of optimal number of processors can be extended to any reservoir simulator and any HPC clusters.

Chapter 6

Conclusions and Suggestions

6.1 Conclusions

In this dissertation, the iterative coupling technique is applied to two-phase and three-phase black oil problems, and the iterative models have been developed. Several numerical techniques are also investigated and incorporated into the new models. Numerical experiments based on both single processor simulation and HPC simulation are conducted to show the performance of the iterative model. All of these work in this dissertation can be summarized as following:

- As a newly developed reservoir simulation scheme, the iterative method is more accurate and stable than the IMPES method; it allows a larger time step and produces smaller solution oscillations with high heterogeneous reservoir. It achieves successful results with coning water problem and countercurrent flow problems, where IMPES has difficulties to solve.
- The iterative method is much faster than the FIM with similar accuracy.

For two-phase problems, 30%-40% simulation time is saved with the iterative method. For the black oil model, around 30% simulation time is saved. At the same time, the mass balance errors are as small as 10^{-4}.The iterative method is easy to be implemented due to its simpler numerical structure.

- Extrapolation, a forcing function, and several other numerical techniques are applied to the iterative method. Phase scaling is utilized both in the air-water model and black oil model, which accelerates the iteration convergence and cuts down iteration numbers. All of these techniques have significantly improved the performance of the iterative model without apparently adding simulation time.

- Several linear solvers with various preconditioners are investigated and compared. For single processor simulation, GMRES preconditioned with SAMG produces a fast solution. For HPC simulation, HYPRE shows great advantages over other solvers of two-phase problem; for black oil problem, HYPRE and SAMG take similar times to solve the problems. HYPRE is highly scalable with the iterative model in HPC environments.

- The iterative model is highly scalable and efficient with the domain decomposition methodology and MPI library. The portion of parallel code of the model is 99.4%. It produces more than 70% parallel efficiency with more than 2000 processors. The parallel efficiency is verified on different computing platforms. It maintains the same accuracy and stability on HPC as on single processor simulation.

- Parallel efficiency on HPC relates to several issues including cluster architecture, problem size, solver, and domain decomposition schemes. The strategy

to estimate the optimal match between a given problem size and the number of processors has been investigated in detail, which will save computation resources and shorten the simulation time. The CPU time distribution among different tasks has been analysed and corresponding improvement have been suggested.

6.2 Suggestions

This dissertation covers a variety of topics related to iterative coupling, and there are many possible extensions could be discussed in following work. Some of these possibilities are listed below:

- Extend iterative techniques to a compositional model that is more complicated. All the numerical techniques approved efficient within the iterative black oil model can be extended to compositional model under certain modification. The development of an iterative compositional model can be started from the compositional IMPEC model within current IPARS simulation tool.

- More advanced linear solvers can be applied to the iterative model to enhance its performance, especially those linear solvers that are designed for parallel computation. These linear solvers can be efficiently coupled based on an analysis of their numerical characters. Optimal parameters of a particular linear solver need to be adopted and verified.

- Iterative techniques can be used to couple multi-model and multi-physics simulations into one framework. The idea is to divide the big reservoir domain into many subdomains; then apply different models to those subdomains based

on their particular physical situations. That is extremely important for super-large, high heterogeneous reservoir problem. The iterative method is a good candidate to solve the interface problem.

- Several finite element methods can be incorporated into IPARS with the iterative method.For example, the Discontinuous Galerkin method is good in computing flow equations and the mixed finite element method is suitable for solving transport equations. Additionally, both methods are proved locally conservative. This kind of coupling enables the simulator to handle large size complicated problems.

Nomenclature

α phases (w, o, g)

ϕ porosity (fraction)

γ constant close to 1, in forcing function

η forcing term applied to linear tolerance

λ_α mobility of phase α

μ fluid viscosity

ρ density

ρ_{ref} reference density

ψ effective mass flux

Λ mass mobility

Λ_t total mass mobility

B formation volume factor

C compressibility

$D(S_w)$ diffusion function in Kirchhoff transformation

f_w fractional flow of water

G dimensionless geometric factor

g gravity constant

K reservoir permeability

k kth iteration in time step $n + 1$

$k_{r\alpha}$ relative permeability

$k_{x,y}$ reservoir permeability in x and y directions

$\overline{k}$ average permeability

L length of open wellbore penetrating

M molecular weight of air

N_M component concentration

P_α pressure in phase α

P_{caw} air-water capillary pressure

P_{cgo} gas-oil capillary pressure

P_{cow} oil-water capillary pressure

P_{wb} bottom-hole pressure in well

q_{MS} mass velocity of component M under standard condition

q_α mass velocity of phase α

R air constant

R_{so} solution gas-oil ratio

r_{eq} equivalent radius of grid cell

r_w wellbore radius

S_α saturation of phase α

T temperature

T_α transmissibility of phase α

t simulation time

U_α mass velocity of phase α

u_α volumetric velocity of phase α

$\triangle x, \triangle y$ dimension of grid cell

Z reservoir depth

$Z(P_a)$ compressibility factor for air phase

Appendix I

The simulation procedure of a general time step of the iterative oil/water model listed following:

- GSAVE: subroutine saves old time step values, at the beginning of each iteration, values of n-1 step are also saved.
- GEXTRAP: if $NStep > 2$, this routine is called only in the first iteration of the current time step. Gets values S_w^{n+1} and P_w^{n+1} as the initial guesses for pressure solve. And then it calls GPROP to update the fluid properties with the extrapolated values, including ρ_o^{n+1}, ρ_w^{n+1}, $Cwat$, $Coil$, and total fluid mass in the reservoir.
- FORCING FUNCTION: uses the forcing function to change the linear tolerance of a pressure solve; the algorithm is shown in section 2.4.2.
- GMOBIL: calculates mobilities for current time step λ^{n+1}.
- GPWMAT: forms a pressure coefficient matrix and residuals array.
- GWELSUMS & GWELLPW: evaluate well contributions to the pressure coefficient matrix; densities are multiplied correspondingly in subroutine GWELLPW to match the new pressure equation.
- Calls a linear solver to solve the pressure equation.
- GUPPRES: updates pressure values; uses new pressure value to update water density and oil density $\rho_w^{n+1}, \rho_o^{n+1}$.
- GVEL: computes mass velocities with new densities.

- GSWMAT: forms saturation matrix and residuals.
- GWELSUMS & GWELLSW: evaluate well contributions to the saturation equation, densities are multiplied correspondingly in subroutine GWELLSW.
- Calls a linear solver to solve the saturation equation.
- GUPSATU: update saturation S_w^{n+1}.
- GPROP: update relative permeability, and P_{cow}, densities, residuals and fluid mass in reservoir.
- Checks convergence of both pressure and saturation solves; calculates oil, water and total mass errors; if it converges, goes to next time step, $NStep = NStep+1$; otherwise, go to next iteration, $Itns = Itns + 1$.
- Recalculates well performance with new properties (GWELLSUMS, GCORRECTWELLS).
- GWELLOUTPUT: updates well data and accumulated injection/production fluid; prints out information.
- Calculates water and oil mass balance $OILBT$, $WATBT$.

The details may be reviewed in the source code of this iterative oil/water model in the IPARS package. The procedures of the iterative air/water model and the iterative black oil model are similar.

Bibliography

[1] Abou-Kassem, J.H., Farouq, S.M.: A unified approach to the solution of reservoir simulation equations. presented at the *SPE Eastern Regional Meeting*, Pittsburgh, PA, Oct. 1987. SPE 17072.

[2] Agarwal, B. and Blunt, M.J.: Full-physics, streamline-based method for history matching performance data of a north sea field. presented at the *SPE Reservoir Simulation Symposium*, Houston, TX, Feb. 2001. SPE 66388.

[3] Agarwal, B. and Blunt, M.J.: Streamline-based method with full-physics forward simulation for history-matching performance data of a north sea field. presented at the *2001 SPE Symposium on Reservoir Simulation*, Houston, TX. Feb. 2001. SPE 84592.

[4] Arbogast, T. and Wheeler, M.F.: A characteristics-mixed finite element method for advection-dominated transport problems. *SIAM J. Numer. Anal.*, 32(2):404 424, 1995.

[5] Arbogast, T., Dawson, C.N., Keenan, P.T., Wheeler, M.F. and Yotov, I.: Enhanced cell-centered finite differences for elliptic equations on general geometry. *SIAM J. Sci. Comp.*, 19(2):404-425, 1998.

[6] Aziz, K. and Settari, A.: Petroleum Reservoir Simulation. *Applied Science Pub.*, London, 1979.

[7] Bank, R.E.: A domain decomposition for a parallel adaptive meshing algorithm. *SIAM J. Sci. Comput.* 22(2000), pp. 1411-1443.

[8] Bastian, P. and Riviere, B.: Discontinuous Galerkin methods for two-phase flow in porous media. *Technical Report*, 2004-28, IWR, SFB 359, 2004.

[9] Behie, G.A., Forsyth, P.A. and Sammon, P.H.: Adaptive implicit methods applied to thermal simulation. *SPE Reservoir Engineering*, Nov. 1987. SPE 14043.

[10] Berenblyun, R.A., Shapiro, A.A., Jessen, K. and Orr, F.M.: Black oil streamline simulator with capillary effects. presented at the *SPE Annual Technical Conference and Exhibition*, Denver, CO. Oct. 2003. SPE 84037.

[11] Berfamaschi, L., Mantica, S. and Manzini, G.: A mixed finite element-finite volume formulation of the black-oil model. *SIAM J. SCI. COMPUT.* Vol.20, No.3, pp.970-997. 1998.

[12] Brand, C.W. and Heinemann, Z.E.: A new iterative solution technique for reservoir simulation equations on locally refined grids. presented at the *1989 SPE Symposium on Reservoir Simulation*, Houston, TX. Feb. 1989. SPE 18410.

[13] Breit, V.S., Bishop, K.A., Green, D.W. and Trompeter, E.E.: A technique for assessing and improving the quality of reservoir parameter estimates used in numerical simulators. presented at the *48th Annual Fall Meeting of the Society of Petroleum Engineers of AIME*, LV, NV. Sep. 1973. SPE 4546.

[14] Buchwalter, J.L. and Miller, C.A.: A new simplified compositional simulator.

presented at *SPE Rocky Mountain Regional/Low Permeability reservoir symposium*, Denver, CO, Apr. 1993. SPE 25858.

[15] Cao, H. and Aziz, K.: Performance of IMPSAT and IMPSAT-AIM models in compositional simulation. presented at *SPE Annual Technical conference and Exhibition*, San Antonio, Texas, Sep. 2002. SPE 77720.

[16] Chen, Z., Huan, G. and Ma, Y.: Computational methods for multiphase flows in porous media. *SIAM Comp. Sci. Eng.*, Philadelphia, 2006.

[17] Chen, Z., Huan, G. and Li, B.: An improved IMPES method for two-phase flow in porous media. *Transport in Porous Meida*, 54: 361-376, 2004.

[18] Chen, Z.: Formulations and numerical methods of the black oil model in porous media. *SIAM J. NUMER. ANAL.* Vol.38, No.2, pp.489-514., 2000.

[19] Chen, Z., Qin, G. and Ewing, R.E.: Analysis of a compositional model for fluid flow in porous media. *SIAM J. APPL. MATH.*, Vol.60, No.3, pp.747-777. 2000.

[20] Cheng, H., Osako, I. and D-Gupta A.: A rigorous compressible streamline formulation for two- and three-phase black oil simulation. presented at the *SPE Annual Technical Conference and Exhibition*, Dallas, TX, Oct. 2005. SPE 96866.

[21] Choo, Y.K. and Welch, V.S.: A comprehensive black oil reservoir simulator for microcomputer applications. In the *Petroleum Industry Applications of Microcomputers*, Montgomery, TX, Jun. 1987. SPE16490.

[22] Christie, M.A. and Blunt, M.J.: Tenth SPE comparative solution project: A comparison of upscaling techniques. presented at the *SPE Reservoir Simulation Symposium*, Houston, TX. Apr. 2001. SPE 66599.

[23] Coats, K.H.: Reservoir simulation: State of art. *JPT*, 1633-1642, Aug. 1982. SPE 10200.

[24] Coats, K.H., Thomas, L.K. and Pierson, R.G.: Compositional and black oil reservoir simulation. 1995. SPE 29111.

[25] Coats, K.H.: A note on IMPES and some IMPES-based simulation models. presented at the *15th symposium on reservoir simulation*, Houston, TX, Feb. 1999. SPE 49774.

[26] Coats, K.H.: IMPES stability: the CFL limit. presented at the *SPE Reservoir Simulation Symposium*, Houston, TX. Mar. 2001. SPE 85956.

[27] Coats, K.H.: IMPES stability: selection of stable time steps. presented at the *SPE Reservoir Simulation Symposium*, Houston, TX. Feb. 2001. SPE 84924.

[28] Collins, D.A., Nghiem, L.X., Li, Y.K. and Grabenstetter, J.E.: An efficient approach to adaptive implicit compositional simulation with an equation of state. *SPERE* l(4):259-264. 1992.

[29] Cook, R.E., Jacoby, R.H. and Ramesh, A.B.: A beta-type reservoir simulator for approximating compositional effects during gas injection. *SPEJ* 14: 471-481, 1974.

[30] Dawson, C.N., Sun, S. and Wheeler, M.F.: Compatible algorithms for coupled flow and transport. *Computer Methods in Applied Mechanics and Engineering*, 193:2565 2580, 2004.

[31] Dawson, C.N., Klie, H., Wheeler, M.F. and Woodward, C.S.: A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver. *Computational Geosciences*, 215-249, 1997.

[32] Dean, R.H., Gai, X., Stone, C.M. and Minkoff, S.E.: A comparison of techniques for coupling porous flow and geomechanics. presented at *SPE Reservoir Simulation Symposium*, Houston, TX, Feb. 2003. SPE 79709.

[33] Dean, R.: Compositional model with IMPEC formular in IPARS. IPARS Manual version 3.

[34] Debaun, D., Byer, T., Childs, P., Chen, J., Saaf, F., Wells, M., Liu, J., Cao, H., Pianelo, L., Tilakraj, V., Crumpton, P., Walsh, D., Yardumian, H., Zorzynski, R., Lim, K.T., Schrader, M., Zapata, V., Nolen, J. and Tchelepi, H.: An extensible architecture for next generation scalable parallel reservoir simulation. presented at the *SPE Reservoir Simulation Symposium*, Houston, TX, Feb. 2005. SPE 93274.

[35] Dimitrie, B.-C., Bia, P.R. and Sabathier, J.-C.: The "Checker Model", An improvement in modeling naturally fractured reservoir with a tridimensional, triphasic, black oil numerical model. In the *SPE Annual Technical Conference and Exhibition*, New Orleans, Sep. 1982. SPE 10977.

[36] Douglas, J. and Peaceman, D.W.: Numerical solution of two dimensional heat flow problems. *A.I. Ch. E. Jour.*, 1:505-512, 1955.

[37] Eaton, F.J.: A multigrid preconditioner for two-phase flow in porous media. PhD dissertation, University of Texas at Austin, Aug. 2001.

[38] Eclipse User Guide 2004A, Schlumberger, 2004.

[39] Eisenstat, S.C., Elman, H.C. and Schultz, M.H.: Block-preconditioned conjugate-gradient-like methods for numerical reservoir simulation. *SPE Reservoir Engineering*, Feb. 1988. SPE 13534.

[40] Eisenstat, S.C. and Walker, H.F.: Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Sci. Comp.*, 1994.

[41] Eriksson, K., Estep, D., Hansbo, P. and Johnson, C.: Computational differential equations. Cambridge, 1996.

[42] Eslinger, O.W.: Discontinuous Galerkin finite element methods applied to two-phase, air-water flow problem. PhD dissertation, University of Texas at Austin, Aug. 2005.

[43] Fagin, R.G., Stewart, C.H. Jr.: A new approach to the two-dimensional multiphase reservoir simulator, SPEJ 1966, SPE 1188.

[44] Fanchi, J.R.: Phase transitions in black oil reservoir simulation. *SPEJ*, Apr. 1987. SPE 17003.

[45] Farnstrom, K.L. and Ertekin, T.: A versatile, fully implicit, black oil simulator with variable bubble-point option. presented at the *SPE California Regional Meeting*, Ventura, CA. Apr. 1987. SPE 16342.

[46] Fevang, Q., Singh, K. and Whitson, C.: Guidelines for choosing compositional and black oil models for volatile oil and gas condensate reservoir. presented at the *2000 SPE Annual Technical Conference and Exhibition*, Dallas, TX, Oct. 2000. SPE 63087.

[47] Fjerstad, P.A., Sikandar, A.S., Cao, H., Liu, J. and Da Sie, W.: Next generation parallel computing for large-scale reservoir simulation. presented at the *SPE International Improved Oil Recovery Conference in Asia Pacific*, Kuala Lumpur, Malaysia. Dec. 2005. SPE 97358.

[48] Forsythe Jr., P., Sammon, P.H.: Gas phase appearance and disappearance in fully implicit black oil simulation. presented on *SPE Annual Technique Conference and Exhibition*, Oct. 1984. SPE 11757.

[49] Fung L.S., Vollins D.A. and Nghiem L.X.: An adaptive implicit switching criterion based on numerical stability analysis. *SPE Res. Eng.* 45-52, Feb. 1998.

[50] Fung, L.S. and Al-Shaalan, T.M.: Parallel iterative solver for the dual-porosity, dual-permeability system in fractured-reservoir simulation. presented at the *International Petroleum Technology Conference*, Doha, Qatar, Nov. 2005.

[51] Fussell, L.T. and Fussell, D.D.: An iterative technique for compositional reservoir models. *SPEJ*, 211, Aug. 1979.

[52] Gai X.: A coupled geomechanics and reservoir flow model on parallel computers. PhD Dissertation, University of Texas at Austin, Aug. 2004.

[53] Gai, X., Sun, S., Wheeler, M.F. and Klie, H.: A time stepping scheme for coupling reservoir flow and geomechanics on non-matching grids. presented at *SPE Annual Technical Conference and Exhibition*, Dallas, TX. Oct. 2005. SPE 97054.

[54] Gai, X., Dean, R., Wheeler, M.F. and Liu, R.: Coupled geomechanical and reservoir modeling on parallel computers. SPE 79700

[55] Ganzer, L.J. and Heinemann, Z.E.: Reservoir simulation using mixed models. presented at the *14th SPE Symposium on Reservoir Simulation*, Dallas, TX. Jun. 1997. SPE 37981.

[56] Gjerde, O., Jarosch, H.S. and Kaarstad, T.: Improvements to the black-oil simulator. Aug. 1988. SPE 18612.

[57] Hoang, H.M. and Kleppe, J.: A parallel adaptive finite difference algorithm for reservoir simulation. presented at the *SPE Europec/EAGE Annual Conference and Exhibition*, Vienna, Austria. Jun. 2006. SPE 99572.

[58] Hoteit H. and Firoozabadi, A.: Black oil modelling by the combined discontinuous Galerkin and Mixed methods. presented at *SPE Annual Technical Conference and Exhibition*, Houston, TX, Sep. 2004. SPE 90276.

[59] Hoyland, L.A., Papatzacos, P. and Skjaeveland, S.M.: Critical rate for water coning: correlation and analytical solution. *SPE Reservoir Engineering*, Nov. 1989. SPE 15855.

[60] Huan G.: The black oil model for a heavy oil reservoir. presented at *International Meeting on Petroleum Engineering*, Beijing, Mar. 1986. SPE 14853.

[61] Huan, G.R.: A flash black oil model. Presented at the *SPE 1985 Middle East Oil Technical Conference and Exhibition*, Bahrain, Mar. 1985. SPE 13521.

[62] HYPRE: high performance preconditioners, user's manual version 2.0.0. Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Dec. 2006.

[63] IBM Rochester Blue Gene Capacity on Demand Centre, User Guide.

[64] Jenkins, E.W.: The IPARSv2 air-water model. TICAM report 2002. The University of Texas at Austin.

[65] Ji, L., Settari, A. and Sullivan, R.B.: A new approach to hydraulic fracturing modelling - fully coupled with geomechanical and reservoir simulation. presented at the *SPE Europec/EAGE Annual Conference and Exhibition*, Vienna, Austria. Jun. 2006. SPE 99428.

[66] Kazemi, H., Vestal, C.R., and Shank, G.D.: An efficient multicomponent numerical simulator. *SPEJ*, 355-68, Oct. 1978.

[67] Kendall, R.P., Morrell, G.O., Peaceman, D.W., Silliman, W.J., and Watts, J.W.: Development of a multiple application reservoir simulator for use on a vector computer. presented at the *SPE Middle East oil Technical Conference*, Manama, Bahrain, Mar. 1983. SPE 11483.

[68] Killough, J.E.: Ninth SPE comparative solution project: a reexamination of black-oil simulation. presented at the *13th SPE Symposium on Reservoir Simulation*, San Antonio, TX. Feb. 1995. SPE 29110.

[69] Killough, J.E.: Is parallel computing ready for reservoir simulation? A critical analysis of the state of the art. presented at the *68th Annual Technical Conference and Exhibition*, Houston, TX, Oct. 1993. SPE 26634.

[70] Killough, J.E. and Wheeler, M.F.: Parallel iterative linear equation solvers: An investigation of domain decomposition algorithms for reservoir simulation. *1987 SPE Reservoir Simulation Symposium*, San Antonio, TX. Feb 1987. SPE 16021

[71] Klie H.: Krylov-secant methods for solving large scale systems of coupled nonlinear parabolic equations. PhD dissertation, Rice University, 1996.

[72] Klie, H. and Wheeler, M.F.: Krylov-Secant methods for accelerating the solution of fully implicit formulations. presented at the 2005 *SPE Reservoir Simulation Symposium*, Houston, TX. Jan. 2005. SPE 92863.

[73] Kwak, D.Y.: V-cycle multigrid for cell-centered finite differences. *SIAM J. SCI. COMPUT.*, Vol.21, No.2, pp.552-564. 1999.

[74] Kwak, D.Y.: V-cycle multigrid convergence for cell centered finite difference method, 3D case. *Fourteenth International Conference on Domain Decomposition Methods*, 2003.

[75] Lacroix, S., Vassilevski, Y.V. and Wheeler, M.F.: Iterative solvers of the implicit parallel accurate reservoir simulator(IPARS), I: single processor case. TICAM report 2002. The University of Texas at Austin.

[76] Lacroix, S., Vassilevski, Y.V., Wheeler, J.A. and Wheeler, M.F.: Iterative solution methods for modeling multiphase flow in porous media fully implicitly. *SIAM J. SCI. COMPUT.*, Vol.25, No.3, pp.905-926. 2003.

[77] Lu, Q., Peszynska, M. and Gai, X.: Implicit black oil model in IPARS framework IPARSv2, Model I, Keyword BLACKI. *TICAM report*, Oct. 2001.

[78] Lu, Q.: A parallel multiblock/multiphysics approach for multiphase flow in porous media. PhD dissertation, University of Texas at Austin, May 2000.

[79] Lu, Q., Peszynska, M. and Wheeler, M.F.: A parallel multiblock black-oil model in multimodel implementation. presented at the *2001 SPE Reservoir Simulation Symposium*, Houston, TX. Feb. 2001. SPE 79535

[80] Lu, B., Al-Shaalan, T.M. and Wheeler, M.F.: Iteratively coupled reservoir simulation for multiphase flow. presented at the *SPE Annual Technical Conference and Exhibition*, Anaheim, CA. Nov. 2007. SPE 110114

[81] Ma, Z.Y., Jing, F.J., Xu, X.M. and Sun, J.C.: Simulation of black oil reservoir on distributed memory parallel computers and workstation cluster. presented at the *International Meeting on Petroleum Engineering*, Beijing, Nov. 1995. SPE 29937.

[82] Magras, J.F., Quandalle, P. and Bia, P.: High-performance reservoir simulation with parallel ATHOS. presented at the *SPE Reservoir Simulation Symposium*, Houston, TX, Feb. 2001. SPE 66342.

[83] Mantica, S., Cominelli, A. and Mantica, G.: Combining global and local optimisation techniques for automatic history matching production and seismic data. presented at the *2001 SPE Reservoir Simulation Symposium*, Houston, TX. Feb. 2001. SPE 78353.

[84] Mungan, N.: A theoretical and experimental coning study. presented at *SPE-AIME 49th Annual Fall Meeting*, Houston, TX. Oct. 1974. SPE 4982.

[85] Norvik, H.: Fluid parameters for black oil simulators. *JPT*, May 1985. SPE 14700.

[86] Peaceman, D.W.: Fundamentals of Numerical Reservoir Simulation. Elsevier Scientific Publishing Co., Amsterdam, 1977.

[87] Peaceman, D.W. and Rachford, H.H.: The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Appl. Math.* 3:28-41, 1995.

[88] Peszynska, M., Lu, Q. and Wheeler, M.F.: Coupling different Numerical algorithms for two phase fluid flow. In J.R. Whiteman, editor, *MAFELAP Proceedings of Mathematics of Finite Elements and Applications*, 205-214, Brunel University, Uxbridge, U.K., 1999.

[89] Peszynska, M.: Multiphysics coupling of three-phase and two-phase models of flow in porous media. in *Analysis and simulation of multifield problems*. Lecture Notes in Applied abd Computational Mechanics 12. Springer-verlag, 2003.

[90] Peszynska, M., Jenkins, E.W. and Wheeler, M.F.: Boundary conditions for fully implicit two-phase flow models. *Contemporary Mathematics* Vol.306, 2000.

[91] Peters, E.J.: Lecture of Advanced Petrophysics. The University of Texas at Austin, 2003.

[92] The Portland Group, PGI User's Guide: *Parallel Fortran, C and C++ for Scientists and Engineers*. 2005.

[93] Qian, E., Espedal, M.S., Garrido, I. and Fladmark, G.E.: Parallel simulation of multiphase/multicomponent flow models. Lecture Notes in Computational Science and Engineering. 2005 Vol. 40, pages 99-114.

[94] Quandalle, P. and Savary, D.: An implicit pressure and saturation approach to fully black oil simulation. presented at *1989 Symposium on Reservoir Simulation*, Houston, TX, Feb. 1989. SPE 18423.

[95] Riviere, B. and Wheeler, M.F.: Locally conservative algorithms for flow. In *MAFELAP Proceedings of Mathematics of Finite Elements and Applications*, 29-46, 1999.

[96] Rubin, B. and Blunt, M.J.: Higher-order implicit flux limiting schemes for black oil simulation. presented at the *11th SPE Symposium on Reservoir Simulation*, Anaheim, CA. Feb. 1991. SPE 21222.

[97] Russel, T.F.: Stability analysis and switching criteria for adaptive implicit methods based on the CFL condition. presented at the *SPE Symposium on Reservoir Simulation*, Houston, TX, Feb. 1989. SPE 18416.

[98] Russel, T.F. and Wheeler, M.F.: Finite element and finite difference methods

for continuous flows in porous media. *The Mathematics of Reservoir Simulation*, R.E. Ewing, ed., 35(106), 1983.

[99] SAMG User's Manual Release 22c, Fraunhofer Institute Algorithms and Scientific Computing, Jun. 2006.

[100] SAMGp User's Manual Release 21z, Fraunhofer Institute Algorithms and Scientific Computing, Oct. 2005.

[101] Settari, A. and Aziz, K.H.: A computer model for two-phase coning simulation. *SPEJ* Jun. 1974. SPE 4285.

[102] Shaw, D.C.: The treatment of wells, faults, and other singularities in a black-oil, finite-element reservoir simulator. presented at the *12th SPE Symposium on Reservoir Simulation*, New Orleans, LA. Feb 1993. SPE 25246.

[103] Shenawi, S.H., Wu, C.H., Jiang, C. and Luan, Z.A.: A new iterative mathematical model for the analysis of imbibition carbonated waterflood in naturally fractured reservoirs. presented at the *1994 SPE Permian Basin Oil and Gas Recovery Conference*, Midland, TX. Mar. 1994. SPE 27717.

[104] Shiralkar, G.S., Fleming, G.C., Watts, J.W., Wong, T.W., Coats, B.K., Mossbarger, R., Robbana, E. and Batten, A.H.: Development and field application of a high performance, unstructured simulator with parallel capability. presented at the *SPE Reservoir Simulation Symposium*, Houston, TX, Feb. 2005. SPE 93080.

[105] Spillette, A.G., Hillestad, J.G. and Stone, H.L.: A high-stability sequential solution approach to reservoir simulation. *48th Annual Technical Meeting*, Las Vegas, Nevada, Oct. 1973. SPE 4542.

[106] Spivak, A. and Dixon, T.N.: Simulation of gas-condensate reservoir. *SPEJ* Dec. 1974. pp.619-632.

[107] Stuben, K., Clees, T., Klie, H., Lu, B. and Wheeler, M.F.: Algebraic multigrid methods(AMG) for the efficient solution of fully implicit formulations in reservoir simulation. Presented at the *SPE reservoir simulation symposium*, Houston, TX, 2007. SPE 105832.

[108] TACC, Lonestar User's Guide, www.tacc.utexas.edu/services/userguides/lonestar/

[109] Tan, T.B. and Kaiogerakis, N.: A fully implicit, three-dimensional, three-phase simulator with automatic history-matching capability. presented at the *11th SPE Symposium on Reservoir Simulation*, Anaheim, CA. Feb. 1991. SPE 21205.

[110] Thomas, G.W. and Thurnau, D.H.: The mathematical basis of adaptive implicit method. presented at the *6th SPE Symposium on Reservoir Simulation*, Dallas, TX, Feb. 1982. SPE 10495.

[111] Tran, D., Nghiem, L. and Buchanan, L.: Improved iterative coupling of geomechanics with reservoir simulation. presented at the *2005 SPE Reservoir Simulation Symposium*, Houston, TX. Feb. 2005. SPE 93224.

[112] Tran, D., Nghiem, L. and Buchanan, L.: An overview of iterative coupling between geomechanical deformation and reservoir flow. presented at 2005 *SPE International Thermal Operations and Heavy Oil Symposium*. Calgary, Canada. Nov. 2005. SPE 97879.

[113] Trangenstein, J.A. and Bell, J.B.: Mathematical structure of the black

oil model for petroleum reservoir simulation. *SIAM Journal on App. Math.*, 49(3):749-783, 1989.

[114] Trimble, R.H. and McDonald, A.E.: A strongly coupled, fully implicit, three-dimensional, three-phase well coning model. *SPE-AIME*, pp.454-459. 1981.

[115] Vassilevski, Y.V.: Iterative solvers of the implicit parallel accurate reservoir simulator(IPARS), II: parallelization issues. TICAM report 2002. The University of Texas at Austin.

[116] Wan, J., Sarma, P., Usadi, A.K. and Beckner, B.L.: General stability criteria for compositional and black-oil models. presented at the *2005 SPE Reservoir Simulation Symposium*, Houston, TX. Jan. 2005. SPE 93096.

[117] Watts, J.W.: A compositional formulation of the pressure and saturation equations. 1985. SPE 12244.

[118] Watts, J.W., and Shaw, J.S.: A new method for solving the implicit reservoir simulation matrix equation. presented at *2005 SPE Reservoir Simulation Symposium*, Houston, Jan. 2005. SPE 93068.

[119] Watts, J.W.: A conjugate gradient-truncated direct method for the iterative solution of the reservoir simulation pressure equation. *Society of Petroleum Engineers of AIME*, Jun. 1981. SPE 8252.

[120] Watts, J.W.: Reservoir simulation: past, present and future. presented on *1997 SPE Reservoir Simulation Symposium*, Dallas, TX. Jun. 1997. SPE 38441.

[121] Weinstein, H.G., Chappelear, J.E. and Nolen, J.S.: Second comparative solution project: a three-phase coning study. *Journal of Petroleum Technology*, Mar. 1986. SPE 10489.

[122] Wheeler, J.A.: IPARS User's Manual. 1995.

[123] Wheeler, M.F. and Yotov, I.: Physical and computational domain decompositions for modelling subsurface flow. In Jan Mandel *et al.*, editors, *Tenth International Conference on Domain Decomposition Methods*, Contemporary Mathematics, 218: 217-228. *American Mathematical Society*, 1998.

[124] Whitson, C.H., Silva, F. and Soride I.: Simplified compositional formulation for modified black oil simulators. presented at *63rd Annual Technical Conference and Exhibition*, Houston, Oct. 1988. SPE 18315.

[125] Wong, T.W., Nutakki, R. and Aziz, K.: A comparison of two approaches to compositional and black oil simulation. presented on *9th SPE symposium on reservoir simulation*, San Antonio, TX, Feb. 1987. SPE 15999.

[126] Wong, T., Firoozabadi, A., and Aziz, K.: The relationship of the Volume Balance method to the Newton-Raphson Method. *SPERE* 415; *Trans. AIME* 289, Aug. 1990.

[127] Young, L.C. and Stephenson, R.E.: A generalized compositional approach for reservoir simulation. *SPEJ*, 727-42, Oct. 1983.

[128] Young, L.C. and Russell, T.F.: Implementation of an adaptive implicit method. *12th Symposium on Reservoir simulation*, New Orleans, LA, Feb. 1993. SPE 25245.

[129] Young, L.C. and Hemanth-Kumar, K.: High-performance black oil computations. presented at the *11th SPE Symposium on Reservoir Simulation*, Anaheim, CA. Feb. 1991. SPE 21215.

Vita

Bo Lu, the son of Wencai Lu and Chengfang Tian, was born in Binzhou, China on May 19, 1979. In April, 2003, he married Ping Zhao in Beijing. He graduated with a Bachelor of Science in Marine Geology from China Ocean University, Qingdao, China, in July 2000. He received a Master of Engineering degree in Petroleum Geology from China University of Petroleum, Beijing, May 2003. In August 2003, he enrolled the graduate school at the University of Texas at Austin to pursue his Ph.D. in the Department of Petroleum and Geosystems Engineering.

Permanent Address: 1622 West 6th ST, Apt L

Austin, Texas 78703

This dissertation was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.